Multiresolution Image Segmentation

DISSERTATION

zur Erlangung des akademischen Grades
doctor rerum naturalium
(Dr. rer. nat.)
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät II
Humboldt-Universität zu Berlin

von
Herr M.Sc. Mohammed Abdel-Megeed M. Salem
geboren am 23.02.1976 in Kairo, Ägypten

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Dr. h.c. Christoph Markschies

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät II:
Prof. Dr. Wolfgang Coy

Gutachter:

1. Prof. Dr.-Ing. Beate Meffert
2. Prof. Dr.-Ing. Ulrich Rückert
3. Prof. Dr. Mohamed F. Tolba

eingereicht am: 26. August 2008
Tag der Verteidigung: 21. November 2008

„Gedruckt mit Unterstützung
des Deutschen Akademischen Austauschdiensts“

Abstract

More and more computer vision systems take part in the automation of various applications. The main task of such systems is to automate the process of visual recognition and to extract relevant information from the images or image sequences acquired or produced by such applications. One essential and critical component in almost every computer vision system is image segmentation. The quality of the segmentation determines to a great extent the quality of the final results of the vision system.

New algorithms for image and video segmentation based on the multiresolution analysis and the wavelet transform are proposed. The concept of multiresolution is explained as existing independently of the wavelet transform. The wavelet transform is extended to two and three dimensions to allow image and video processing. The investigation of various Daubechies wavelets shows that the Haar wavelet is the best suited wavelet for the proposed algorithms and the investigated applications.

For still image segmentation the Resolution Mosaic Expectation Maximization (RM-EM) algorithm is proposed. The principle of this algorithm is that the conventional EM algorithm is applied to a resolution mosaic of the image as a kind of pre-processing. The resolution mosaic enables the algorithm to employ the spatial correlation between the pixels. The level of the local resolution depends on the information content of the individual parts of the image. The use of various resolutions speeds up the processing and improves the results.

New algorithms based on the 3D wavelet transform and the 3D wavelet packet analysis are proposed for extracting moving objects from image sequences. The new algorithms have the advantage of considering the relevant spatial as well as temporal information of the movement. Fast motions are detected better in the first analysis levels whereas slow motions or motions of big objects in the deeper layers. That is why a combination of different levels gives the best results.

Because of the low computational complexity of the wavelet transform an FPGA hardware for the primary segmentation step was designed.

Actual applications are used to investigate and evaluate all algorithms: the segmentation of magnetic resonance images of the human brain and the detection of moving objects in image sequences of traffic scenes. All results are compared with others obtained from published work. The new algorithms show robustness against noise and changing ambient conditions and gave better segmentation results.

Keywords:
Image segmentation, Multiresolution, 3D wavelet transform, FPGA, Traffic monitoring

Zusammenfassung

Systeme der Computer Vision spielen in der Automatisierung vieler Prozesse eine wichtige Rolle. Die wichtigste Aufgabe solcher Systeme ist die Automatisierung des visuellen Erkennungsprozesses und die Extraktion der relevanten Information aus Bildern oder Bildsequenzen. Eine wichtige Komponente dieser Systeme ist die Bildsegmentierung, denn sie bestimmt zu einem großen Teil die Qualität des Gesamtsystems.

Für die Segmentierung von Bildern und Bildsequenzen werden neue Algorithmen vorgeschlagen. Mathematische Grundlage dieser Algorithmen sind die Multiresolutionsanalyse und die Wavelet-Transformation. Das Konzept der Multiresolution wird als eigenständig dargestellt, es existiert unabhängig von der Wavelet-Transformation. Die Wavelet-Transformation wird zur Verarbeitung von Bildern und Bildsequenzen zu einer 2D- bzw. 3D-Wavelet-Transformation erweitert. Die Untersuchung verschiedener Daubechies-Wavelets zeigt, dass das Haar-Wavelet für die vorgeschlagenen Algorithmen und die ausgewählten Anwendungen am besten geeignet ist.

Für die Segmentierung von Bildern wird der Algorithmus *Resolution Mosaic Expectation Maximization* (RM-EM) vorgeschlagen. Das Prinzip dieses Algorithmus ist, dass der konventionelle EM-Algorithmus auf ein vorverarbeitetes Bild angewendet wird. Das Ergebnis der Vorverarbeitung sind unterschiedlich aufgelösten Teilbilder, das Auflösungsmosaik. Durch dieses Mosaik lassen sich räumliche Korrelationen zwischen den Pixeln ausnutzen. Der Grad der Auflösung hängt vom Informationsgehalt der jeweiligen Teilbilder ab. Die Verwendung unterschiedlicher Auflösungen beschleunigt die Verarbeitung und verbessert die Ergebnisse.

Für die Extraktion von bewegten Objekten aus Bildsequenzen werden neue Algorithmen vorgeschlagen, die auf der 3D-Wavelet-Transformation und auf der Analyse mit 3D-Wavelet-Packets beruhen. Die neuen Algorithmen haben den Vorteil, dass sie sowohl die räumlichen als auch die zeitlichen Bewegungsinformationen berücksichtigen. Schnelle Bewegungen werden in den ersten Analysestufen besser detektiert, langsame Bewegungen oder die Bewegung großer Objekte dagegen erst in tieferen Stufen. Deshalb ergeben sich die besten Ergebnisse durch Kombination verschiedener Stufen.

Wegen der geringen Berechnungskomplexität der Wavelet-Transformation ist für den ersten Segmentierungsschritt Hardware auf der Basis von FPGA entworfen worden.

Aktuelle Anwendungen werden genutzt, um die Algorithmen zu evaluieren: die Segmentierung von Magnetresonanzbildern des menschlichen Gehirns und die Detektion von bewegten Objekten in Bildsequenzen von Verkehrsszenen. Die Ergebnisse werden mit denen anderer publizierter Arbeiten verglichen. Die neuen Algorithmen sind robust gegenüber Rauschen und sich ändernden Umgebungsbedingungen und führen zu besseren Segmentierungsergebnissen.

Schlagwörter:
Bildsegmentierung, Mehrfachauflösung, 3D Wavelet-Transform, FPGA, Verkehrsmonitoring

Dedication

To my mother, to my father, to Ahmed and Mostafa

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The problem that is addressed in this thesis is image segmentation for advanced applications using the concept of multiresolution.

The objective is to show that the multiresolution analysis has the ability to decompose the information content of images so that for the segmentation process the relevant information can be utilised and the irrelevant information can be ignored. The idea of multiresolution simplifies the segmentation process and improves its results.

1.1 Motivation

A picture is worth a thousand words. Images are rich in information that needs to be processed and extracted. This can be done not only by humans but nowadays also by machines.

The visual perception may be the most important human perception. It provides consistent and accurate information of the actual state of the environment without being in direct contact with the environment.

In human visual perception a complicated process is behind the interpretation and understanding of an observed scene. The human visual perception is not simply a translation of a retinal stimulus. A percipient does not see a complex scene, but rather a set of objects and relations. Thus, scene segmentation is an essential and critical component [Maj08].

The discipline of computer vision deals with the design of artificial systems that extract information from images. Typical applications - among many others - are quality control in industrial processes, robot systems in manufacturing, computer-human interaction, or visual surveillance.

Typical tasks of such systems are object recognition, event detection and tracking, scene modelling, or image restoration. Similarly to a human vision system, image segmentation is essential in almost every computer vision system and one of the most difficult tasks. It determines the quality of the final results of the system to a great extent.

Segmentation can be defined briefly as the process of partitioning an image into disjoint homogeneous regions. The homogeneity features can be determined better using different resolutions. By selecting proper resolutions suitable discriminating features can be extracted and used for segmenting the image [Toe05].

In addition to the segmentation process the principle of multiresolution is an established part of the human vision system [KPZC06]. It can build different representations of an image with a spatial resolution adapted to the size of objects of interest. An observer describes many scenes in all its particulars for the objects in the foreground and in general terms for the background. The objects in foreground are usually described by lines, shapes, and fine features, while the background is usually described by only one term as an abstract colour or an identifier for a class of objects. If we may use the word *resolution* in conjunction with the word *detail*, the *fine* details are expected to be expressed by an observation with a *high* resolution, the *global* information by a *low* one. We can say that a scene is observed in multiresolution: objects or regions of interest with a high resolution, regions out of interest with a low resolution.

The theory for the multiresolution signal decomposition was first proposed by Mallat in 1989 [Mal89]. Today, the wavelet transform is the most commonly used method to implement the multiresolution representation. Unlike the Fourier transform the wavelet transform is a tool that decomposes signals into components with different resolutions. The signal is splitted into time intervals of different durations. Each time interval is then analysed by the wavelet function in a suitable scale. A certain frequency component can be found in the coefficients of the appropriate scale. Thus, the wavelet transform provides *a tool for simultaneous time and frequency localisation* [Dau92]. Transient features (short-time details) of a signal with jump discontinuities or peaks can be localised easily from looking at the wavelet coefficients that correspond to high frequencies or small scales. Conversely, long-time trends are stored in deeper layers of the coefficient hierarchy and are represented automatically in coefficients that correspond to low frequencies or larger scales. As a consequence, the first kind of coefficients gives good time localisation and bad frequency localisation, the last ones give a good frequency and a bad time localisation.

Images are 2D signals, where the time dimension is replaced by the space dimensions, i.e., rows and columns in the discrete description of the two space dimensions. To apply the wavelet transform for images it has to be extended into a multidimensional transform. This transform is able to extract the spatial frequencies of an image in a similar way as the 1D transform does in time. Fine details as lines or edges or abrupt spatial changes can be found in high spatial frequencies or small scales. Low spatial frequencies, on the other hand, represent global information and can be found in larger scales.

Multiresolution image analysis is a successively coarser and coarser approximation of the original signal. This is interpreted as representing the signal by different levels of resolution. The most obvious advantage of a multiresolution representation is that it provides a possibility for reducing the computational costs of various image processing operations. Integrating this concept in the design of image segmentation algorithms should reduce the complexity of the algorithms to make them capable of a high processing performance.

1.2 Problem Statement

The study and evaluation of existing methods for image and video segmentation points out that further research in this field is necessary. To have simultaneously a good segmentation result and a low complexity of an algorithm is almost unachievable. Especially the detection of moving objects is a challenge for real-time processing. Therefore, in this thesis possibilities are considered to improve the segmentation results with simultaneous consideration of the algorithm complexity to allow implementation by hardware.

To achieve this goal the main focus of the thesis comprises four areas: improving the known Expectation Maximization (EM) algorithm for the segmentation of still images, applying and evaluating the wavelet packet analysis and the 3D wavelet transform for moving picture or video segmentation, and finally the hardware implementation of the first segmentation step to speed up all algorithms.

The main concept for all parts of the thesis is to use the idea of multiresolution and the tool of wavelet transform as a basis for the design of the new algorithms. The wavelet transform will be useful because it distributes the information contained in the image in different subbands so that the segmentation process can be simplified.

The wavelet transform as a tool that provides multiresolution is selected to meet two fundamental requirements. They concern the dimensionality and the relevance of the extracted information:

1. The analysis and the processing must use as many dimensions as the input signal to determine the features of the extracted information. For example, for image segmentation the analysis has to take into account the location of the pixels and not only their grey values. The same is true for moving pictures or videos. If the information to be extracted regards the motion, the analysis must be performed in three dimensions, two space dimensions and the time dimension.

2. The degree of the visual details in the image to be analysed shall correspond to the relevance of the information to be extracted. The relevance of information is high for the objects of interest. Therefore, all details of interest shall be analysed with high resolution. Alternatively, the background or pixels inside homogeneous regions shall be analysed with low resolution, since the information expected does not change much in this area. It is of less importance for the segmentation process.

Our investigations contain an evaluation of the new algorithms, analytically in terms of the complexity and empirically in terms of the quality of the results. The possibility of a hardware implementation and a design of an FPGA-based wavelet procedure is to be investigated too.

There are two different preliminary considerations for the choice of the potential applications. The first is the complexity of the structure of the chosen example and the lack of adequate segmentation methods. The second one is the determination of recent segmentation problems with the goal to find algorithms that can be generalised beyond the specific tasks under consideration.

The appropriate image segmentation is one of the fundamental problems of various neuro-imaging techniques. In the case of Magnetic Resonance Imaging (MRI), for example, it is necessary to investigate the brain dynamics according to different behavioural and stimulation parameters. This information extraction requires efficient and reliable detection and tracing of the corresponding activated brain regions. Additionally, the applications of the MRI attract more and more attention because of the rapid development and distribution of the imaging systems.

In general, image segmentation is needed in many biomedical applications such as organ-model reconstruction and visualisation and assistant systems for the purposes of diagnostics and therapy. So MR images are chosen as a representative example for still image segmentation.

The second application area is the video surveillance. Many outdoor scenes for traffic monitoring are used from various places in Berlin and Stuttgart. In this application area the task of segmentation is the task of separating the areas with moving objects from the background. Outdoor images increase the complexity of the segmentation, since they suffer from inherent problems such as inhomogeneity of the background and the change of ambient conditions.

The relation between the individual components of the dissertation is illustrated in Fig. 1.1.

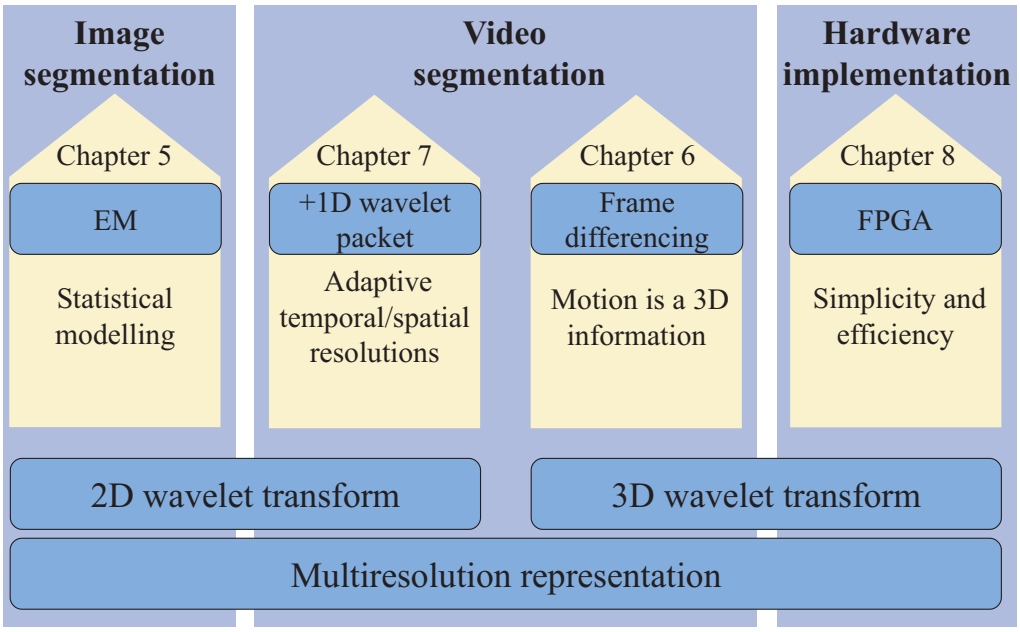


Figure 1.1: Relations between the individual chapters.

1.3 Thesis Outline

The thesis is organised in 10 chapters. The next three chapters focus on the technical and the theoretical backgrounds. The following four chapters are devoted to the proposed segmentation algorithms and the hardware implementation. Then the results are presented and discussed in detail and finally the conclusions are given.

In Chapter 2 the operation of image segmentation is defined. Then a survey of known approaches used for image and video segmentation is given. Based on this survey, the motivation is formulated for the choice of the theoretical basis and the techniques for the thesis.

In Chapter 3 the multiresolution analysis of images is introduced as a tool independent of the wavelet transform. Different techniques used to compute the multiresolution analysis and earlier applications are described at the end of this chapter.

In Chapter 4 the theoretical basis is completed by introducing the wavelet transform. The focus of this chapter is on the dyadic wavelet transform. Definitions of fundamental topics are given, such as the Fourier transform and the continuous wavelet transform. The chapter also includes other related topics, such as the multidimensional wavelet transform and the wavelet packet analysis. At the end of the chapter some recent applications of the wavelet transform in image and video segmentation are described.

In Chapter 5 a new image segmentation algorithm is proposed based on the resolution mosaic and the well-known expectation maximization algorithm. Dependent on the distribution of the information contained in the image, the multiresolution analysis is used to re-represent the image in a mosaic of different resolutions. The expectation maximization algorithm is used to find the missing parameters of the used model.

In Chapter 6 a new algorithm is proposed that allows to derive a motion based segmentation for any frame of an image sequence. The algorithm is based on the multiresolution analysis and the classical 3D wavelet transform. An analytical evaluation of the computational resources used by the algorithm is given in the discussion.

The second algorithm presented in Chapter 7 is proposed for traffic monitoring systems. It is based on the resolution mosaic and the 3D wavelet packet analysis. The resolution mosaic is adapted to the view conditions of the scene.

In Chapter 8 a concept is introduced for a hardware implementation of the 3D wavelet transform and the primary segmentation step of the algorithm proposed in Chapter 6. The objective is to benefit from the inherent parallelism property of the wavelet analysis and its computational simplicity.

The results of all proposed algorithms and the algorithms, which were implemented for comparison reasons, are presented in Chapter 9. A description is given for the used data sets and the methods used for comparing and evaluating the results.

Finally, some conclusions presented in Chapter 10 are highlighting the important results. An outlook on future research directions completes the thesis.

Chapter 2

Image and Video Segmentation

In this chapter a survey is presented on the techniques used for image and video segmentation. Image segmentation methods are given in three groups based on image features used by the method. The video segmentation methods are grouped based on the technique used. The advantages and disadvantages of the existing methods are evaluated, and the motivations to develop new techniques with respect to the addressed problems are given.

2.1 Introduction

Digital images and digital videos are pictures and films, respectively, which have been converted into a computer-readable binary format consisting of logical zeros and ones. An image is a still picture that does not change in time, whereas a video evolves in time and generally contains moving and/or changing objects. Digital images or videos are usually obtained by converting continuous signals into a digital format, although "direct digital" systems are becoming more prevalent.

An important feature of digital images and videos is that they are multidimensional signals, i.e., they are functions of more than a single variable. In the classical study of the digital signal processing the signals are usually one-dimensional functions of time. Images however, are functions of two, and perhaps three space dimensions in case of coloured images, whereas a digital video as a function includes a third (or fourth) time dimension as well. The dimension of a signal is the number of coordinates that are required to index a given point in the image. A consequence of this is that digital image processing, and especially digital video processing is quite data intensive, meaning that significant computational and storage resources are required [Bov00].

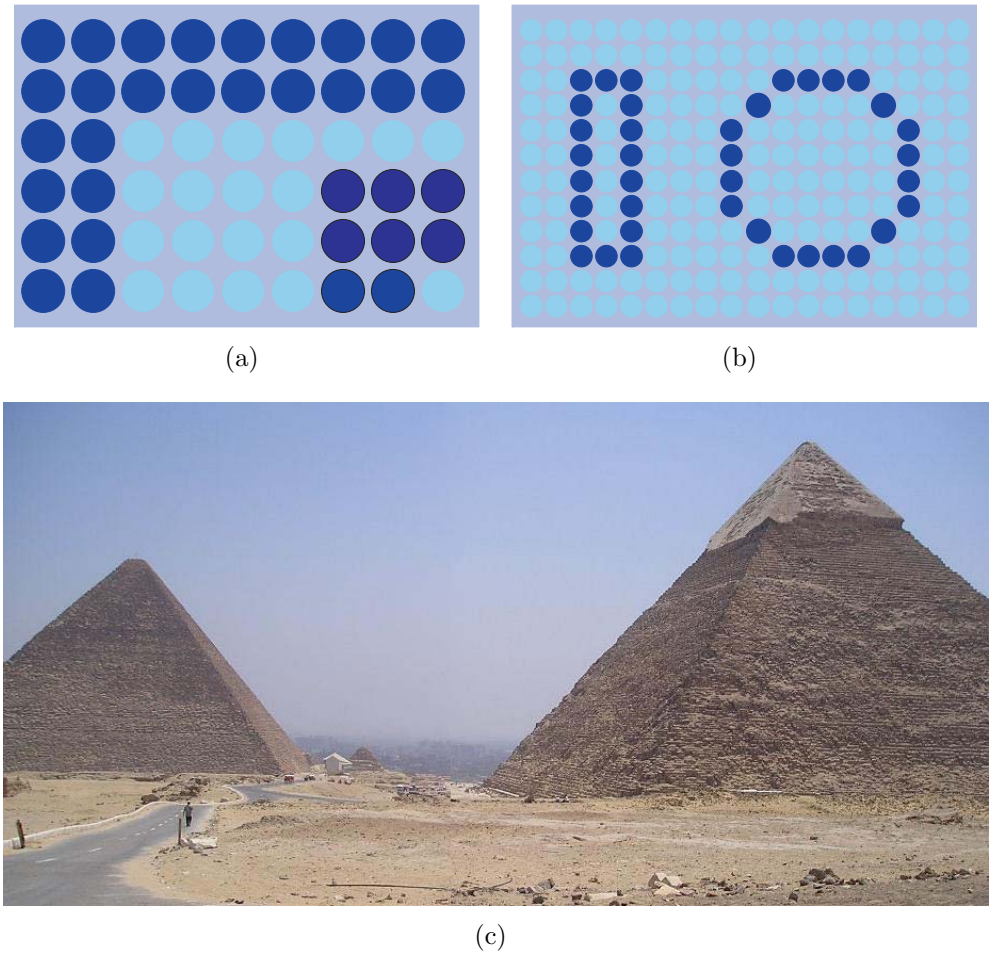
A digital video is obtained either by sampling an analog video signal, or by directly sampling the three-dimensional space-time intensity distribution that is captured by a sensor. In either case, what results is a time sequence of two-dimensional spatial intensity data, or equivalently, a three-dimensional space-time array. Digital videos can be compressed very effectively because of the redundancy inherent in the data and because of an increased understanding of the relationship between the contents of a video stream and the characteristics of the human visual system [Bov00].

2.2 Image Segmentation

2.2.1 Principles

Image segmentation is one of the most important stages in artificial vision systems. It is the first step in almost every pattern recognition process. In some context other terms like *object isolation* or *object extraction* are used.

The human vision system essentially segments the observed scene. One does not see a complex scene, but rather a set of objects. The importance of the process of visual segmentation *grouping* is claimed by the Gestalt theory [Met53] as an early step in the analysis of a visual scene. It states, "whenever points (or previously formed visual objects) have one or more several characteristics in common, they get grouped and form a new larger visual object a *gestalt*" [DMM03]. The psychologists belonging to this school provided a set of guidelines for predicting the process of visual segmentation [TLE02]. Five main guidelines often called *Gestalt laws* have inspired many image segmentation techniques. One of them is the *similarity law*: elements that look more similar are grouped together. Shape or brightness can be meant by the similarity, where bright objects are grouped together and dark objects are grouped together. Fig. 2.1(a) shows an illustration of the similarity law based on the brightness taken from [Met53]. We think that this law has inspired the pixel-based segmentation approaches. The similarity law can be extended to the *self-similarity* of the natural patterns that often happened at multiple spatial scales. Objects can be grouped together if they share the same shape although they are in different scales as shown in Fig. 2.1(c). Another example is the *enclosedness law*: Objects are grouped together if they are arranged on a closed path. In Fig. 2.1(b) two regions can be easily recognised because they are closed inside two different paths. The edge-based and region growing-based segmentation approaches may be inspired from this law.



(a) (b)

(c)

Figure 2.1: Illustration of the Gestalt laws. (a) Similarity law [Met53], (b) Enclosedness law. (c) Self-similarity in multiple scales.

The segmentation is an unconscious activeness by the human observer. However, in the computer vision system it is computationally expensive and a logically non-trivial task.

Image segmentation is computationally the division of an image into disjoint homogeneous regions or classes. All the pixels in the same class must have some common characteristics. The conventional segmentation procedure starts by transforming the original image into a feature space in order to find the boundaries between the different classes. It is followed by a mapping step, which assigns a label to each pixel such that all the pixels of the same features will have the same class. These two steps produce a segmented image.

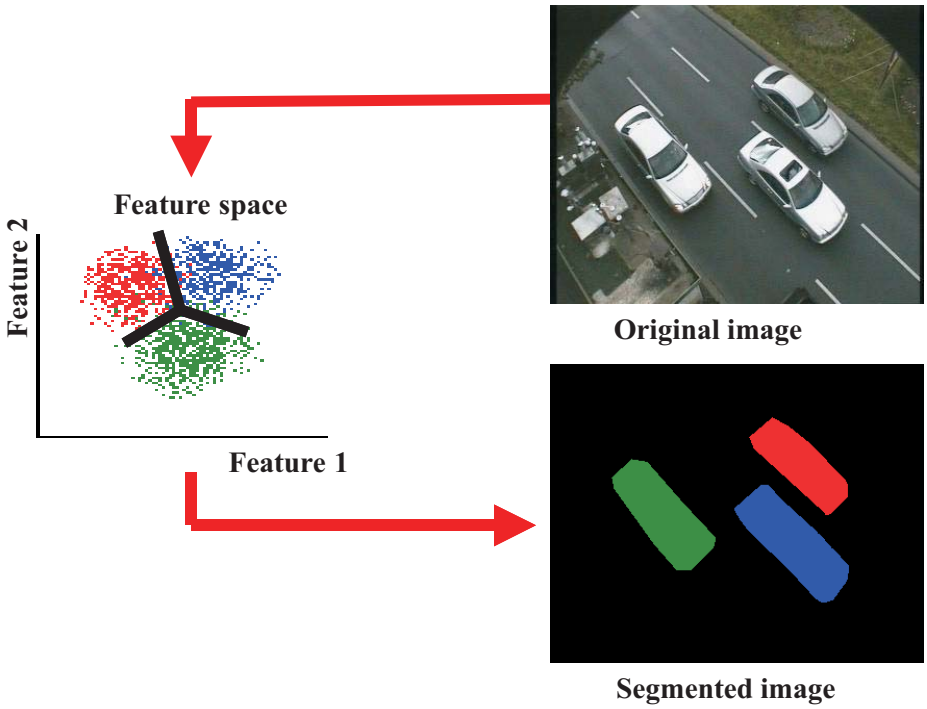


Figure 2.2: Image segmentation.

Consider the image in Fig. 2.2. It is clear that the image consists of three moving objects, street, and street sides. One wishes that all pixels belonging to the moving objects to be assigned to one class, all pixels belonging to the street to another class, and finally all pixels belonging to the street sides to a third class. The question is, how to find the representative features which identify the relationship of the pixels belonging to the same region and how to distinguish them from the pixels belonging to the other regions.The pixel similarity could be measured based on the consistency of, e.g., location, intensity, colour or texture separate or in combination. For example, one can use colour components only or use both location and intensities to create a feature vector for each pixel.

The image segmentation problem can be formulated as following:
The image I can be defined as a function f on the domain in the form of a matrix of $\{(m, n) : 1 \leq m \leq M \text{ and } 1 \leq n \leq N\}$. At any given point (m, n) the value of f can be:

- A binary image (black and white):

$$f(m,n) = \{g : g = 0 \text{ or } 1\}$$

- A grey level image:

$$f(m,n) = \{g : 0 \leq g \leq 255\}$$

- RGB-coloured image:

$$f(m,n) = \{(g_1, g_2, g_3) : 0 \leq g_1 \leq 255, 0 \leq g_2 \leq 255 \text{ and } 0 \leq g_3 \leq 255\}$$

Generally, it can be written as:

$$f : \{(m,n) : 1 \leq m \leq M \text{ and } 1 \leq n \leq N\} \rightarrow \text{colour domain } C \qquad (2.1)$$

A segmented image S is a function s on the same domain as f but it takes its values from different sets. Three common examples are:

- A binary segmentation (foreground and background):

$$s(m,n) = \{k : k = 0 \text{ or } 1\}$$

- A multiclass segmentation (the case of K different segments):

$$s(m,n) = \{k : k = 1, 2, 3, ..., K\}$$

- A boundary image:

$$s(m,n) = \{k : k = 0 \text{ or } 1\}$$

  where 1 could denote the presence of a boundary and 0 the absence.

The image segmentation algorithms can be classified in different categories. In [Hab88, GW93, Cas96] the algorithms are classified either as a simple thresholding method, or grey level edge-based methods, or region-based methods. A similar classification is found in [OH03]. The authors classify the algorithms into three categories: pixel-based segmentation, edge-based segmentation, and region-based segmentation. In [LOPR97] the authors use the same classification with an additional category for texture-based segmentation.

In [ZHT05] the algorithms are classified based on the mathematical tool used into finer classes as: threshold-based methods, clustering methods, region growing methods, edge-based methods, fuzzy-based methods, and neural networks-based methods. In [Toe05] the author classifies the segmentation methods in overlapping classes. First, the methods are classified into location independent methods, e.g., threshold methods, and location dependent methods, e.g., region growing and edge detection methods. Second, the methods are classified based on homogeneity conditions, e.g., region growing, and methods based on discontinuity conditions, e.g., edge detection. Another class is proposed for model-based segmentation. This class overlaps with the other classes in some methods such as region growing methods and interactive edge detection methods. Although the authors in [CS05] discussed only selected methods that are based on edge detection, they have discussed separately the segmentation methods for synthetic images.

Here the segmentation methods are grouped based on the properties of the mapping used to assign classes to the pixels. If the mapping is based on pixel features then the segmentation algorithm is classified under the pixel-based approaches. If the mapping is based on region features then the segmentation algorithm is classified under the region-based approaches. Some methods divide the image into interesting or non-interesting objects or regions based on predefined templates. The regions that are fitting into these templates are considered to be interesting, otherwise non-interesting. These methods are classified under the template-based approaches.

2.2.2 Pixel-based Approaches

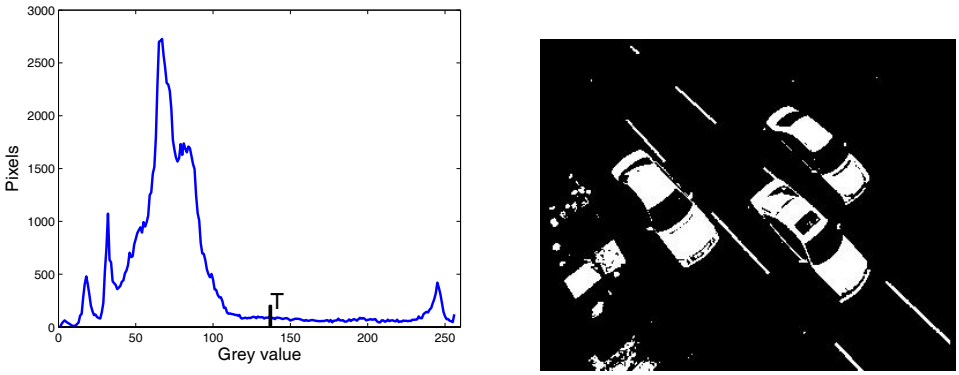
In the pixel-based approaches the properties of single pixels are used to identify the class to which the pixel belongs. The used properties are mainly the pixel intensity or the intensities of the closed neighbourhood of the pixel. The segmentation is done regardless of the position of the pixel in the image or of the characteristics of the structure of the object. That means if two pixels have similar intensities they will be assigned most probably to the same object or class even if they are in separated parts of the image.

Thresholding

Grey level thresholding is one of the oldest, simplest and most popular technique for image segmentation. It can be done based on global information or on local information of the image.

If only one threshold is used for the entire image then we have global thresholding or bi-level thresholding. For that the image is partitioned into two regions (foreground and background). When the image is partitioned into several subregions, it is referred to as local thresholding or as multi-thresholding. In such a situation a set of thresholds $(t_1, t_2, ..., t_k)$ has to be found such that all pixels with a grey level in $[t_i, t_{i+1}), i = 0, 1, 2, ..., k$; constitute the i^{th} region.

If the image is composed of regions with different grey level ranges the histogram of the image usually shows different peaks, each corresponding to one region and adjacent peaks are likely to be separated by a valley. For example, if the image has a distinct object on a background, the grey level histogram as shown in Fig. 2.3, is likely to be bimodal with a deep valley. In this case, the bottom of the valley (T) is taken as the threshold for object/background separation. Therefore, when the histogram has a (or a set of) deep valley(s) thresholding is an appropriate segmentation approach. However, it is not a trivial job [PP93].



(a) (b)

Figure 2.3: Illustration of the thresholding techniques. (a) Histogram of the image in Fig. 2.2 with a global threshold used to separate the objects from the background. (b) Segmentation result.

Many of the thresholding image segmentation methods have a common drawback. They take into account only the histogram information and ignore the spatial details. As a result, such an algorithm may fail to detect thresholds if they are not properly reflected as valleys in the histogram.

The philosophy behind grey level thresholding, that pixels with grey level $\leq T$ fall into one region and the remaining pixels belong to another region, may not be true on many occasions, particularly, when the image is noisy or the background is uneven or the illumination is poor. In such cases, the objects are still lighter or darker than the background, but any fixed threshold level for the entire image will usually fail to separate the objects from the background. This leads to the methods of adaptive thresholding. In adaptive thresholding, normally the image is partitioned into several non-overlapping blocks and a threshold for each block is computed independently. These local thresholds are then interpolated over the entire image to yield a threshold surface [PP93].

A well-known method for image thresholding is the Otsu method [Ots79]. For a grey level image the global threshold value is computed by minimising the variance of the pixel values inside the classes of the foreground and the background and maximising the variance between the two classes.

Let $p(g)$ be the probability function of a grey level g. It is the normalisation value of the histogram over the area of the image at g. Assuming that the background has darker grey levels as the foreground. Then for each possible value of the variable $\tau_{\in\{0..255\}}$, one can define the probability of the class of the background as:

$$P_{BG}(\tau) = \sum_{g=0}^{\tau} p(g) \tag{2.2}$$

and the probability of the class of the foreground as:

$$P_{FG}(\tau) = \sum_{g=\tau+1}^{255} p(g) \tag{2.3}$$

where $P_{BG} + P_{FG} = 1$.

The variance inside the classes $\sigma^2_{in}(\tau)$ can be computed as:

$$\sigma^2_{in}(\tau) = P_{BG}(\tau) \cdot \sigma^2_{BG}(\tau) + P_{FG}(\tau) \cdot \sigma^2_{FG}(\tau) \tag{2.4}$$

where $\sigma^2_{BG}(\tau)$ and $\sigma^2_{FG}(\tau)$ are the variances of the background and foreground classes, respectively.
The variance between the two classes $\sigma^2_{be}(\tau)$ can be computed as:

$$\sigma^2_{be}(\tau) = P_{BG}(\tau) \cdot (\overline{g_{BG}} - \overline{g})^2 + P_{FG}(\tau) \cdot (\overline{g_{FG}} - \overline{g})^2 \tag{2.5}$$

where $\overline{g_{BG}}$, $\overline{g_{FG}}$ and $\overline{g}$ are the mean grey level value of the classes of the background, foreground, and the mean value of the image, respectively.

The algorithm searches afterwards the maximum value of the ratio $Q(\tau)$ of the variance between the classes over the variance inside the classes.

$$Q(\tau) = \frac{\sigma_{be}^2(\tau)}{\sigma_{in}^2(\tau)} \tag{2.6}$$

This implies that the variance inside the classes has to be minimised and the variance between the classes to be maximised. The method of Otsu is used to produce Fig. 2.3.

Other thresholding methods, which use other techniques in combination, are introduced in the following sections.

Statistical Methods

The Expectation Maximization (EM) algorithm was developed and employed independently by several different researchers until Dempsters et al. [DLR97] brought their ideas together, proved convergence, and coined the term "EM algorithm". Since that seminal work hundreds of papers employing the EM algorithm in many areas have been published.

Generally, the EM algorithm produces Maximum Likelihood (ML) estimates of parameters when there is a many-to-one mapping to the distribution governing the observation. The EM algorithm is used widely in the image segmentation field and it produces very good results especially with a limited noise level. The image is considered as a Gaussian mixture model. Each class is represented as a Gaussian model and the pixel intensity is assumed as an observed value of this model. The EM is used for finding the unknown parameters of the mixture model.

A set of observed data $X = \{x_i \mid i = 1, ..., N\}$ can be modelled as to be generated from a mixture of random processes $X_1, X_2, ..., X_K$, with joint probability distribution $f(X_1, X_2, ..., X_K)$, where K is the number of classes or distribution functions present in the mixture. It is usually assumed that these processes represent independent identically distributed random variables. Then one can write:

$$f(X_1, X_2, ..., X_K) = \prod_{k=1}^{K} f(x, \theta_k) \tag{2.7}$$

where $f(x, \theta_k)\ \forall k = 1, 2, ..., K$ is the probability distribution function of the random variable X_k, and $\theta_k = \{\mu_k, \sigma_k\}$ stands for the parameters that define the distribution k.

$$\Phi = \{p_1, ..., p_K, \mu_1, ..., \mu_K, \sigma_1, ..., \sigma_K\}$$

is called the parameter vector of the mixture, where p_k are the mixing proportions ($0 \leq p_k \leq 1, \forall k = 1, ..., K$, and $\sum_k p_k = 1$).

The EM algorithm consists of two major steps: an expectation step (E-step), followed by a maximization step (M-step). The expectation step is to estimate a new mapping (pixel-class membership function) with respect to the unknown underlying variables, using the current estimate of the parameters and conditioned upon the observations. The maximization step then provides a new estimate of the parameters. These steps iterate until convergence is achieved [TM96].

1. The E-Step:

   Compute the expected value of $z_{i,k}$ using the current estimate of the parameter vector Φ as introduced in [Sae97]:

   $$z_{i,k}^{(t)} = \frac{p_k^{(t)} G(x_i \mid \theta_k^{(t)})}{f(x_i \mid \Phi^{(t)})} \tag{2.8}$$

   where $z_{i,k}$ is the probability of x_i belonging to class k, where $1 \leq i \leq N, 1 \leq k \leq K$ and x_i is the intensity value of the pixel i. It should be referenced afterwards as the pixel x_i.
   $z_{i,k}$ satisfies the conditions:
   i) $0 \leq z_{i,k} \leq 1$
   ii) $\sum_k z_{i,k} = 1$
   iii) $\sum_i z_{i,k} > 0$.

   $G(x_i \mid \theta_k^{(t)})$ is the probability of pixel x_i given it is a member of class k.
   p_k is the class proportional in the model $\sum_k p_k = 1$.
   The $f(x_i \mid \Phi)$ is the total probability function that is defined as:

   $$f(x_i \mid \Phi) = \sum_{k=1}^{K} p_k G(x_i \mid \theta_k^{(t)})$$

   The superscript (t) means the iteration number t.

2. The M-Step:

   Use the data from the expectation step as if it were actually measured data and compute the mixture parameters as introduced in [Sae97]:

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{N} z_{i,k}^{(t)} x_i}{\sum_{i=1}^{N} z_{i,k}^{(t)}} \tag{2.9}$$

$$\sigma_k^{2(t+1)} = \frac{\sum_{i=1}^{N} z_{i,k}^{(t)} (x_i - \mu^{(t+1)})^2}{\sum_{i=1}^{N} z_{i,k}^{(t)}} \tag{2.10}$$

$$p_k^{(t)} = \frac{\sum_{i=1}^{N} z_{i,k}^{(t)}}{N} \tag{2.11}$$

The EM algorithm starts with an initial guess $\Phi^{(0)}$ of the parameters of the distributions and the proportions of the distributions in the image. It iterates until a conversion of the parameter vector Φ is achieved. Fig. 2.4 shows its flowchart.

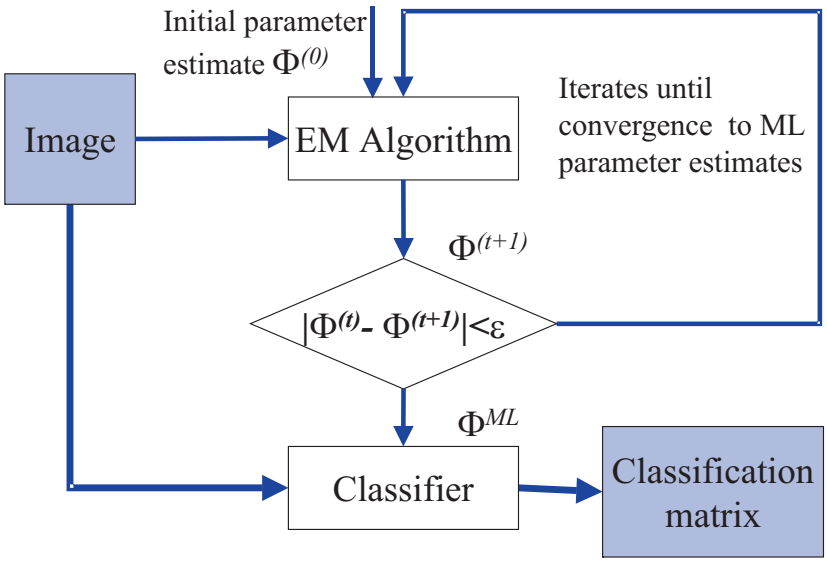


Figure 2.4: Block diagram of the EM algorithm. I: input image. S: segmented image.

The EM algorithm is always followed by a classification step. The EM is producing the missing parameters in Φ, which are then used by a classifier which is defined as:

$$k_i = \underset{k}{\operatorname{argmax}}(G(x_i \mid \theta_k^{(t)})) \tag{2.12}$$

It assigns a class membership to a pixel i depending on its intensity x_i to the class whose parameter vector maximises the Gaussian density function. The value of this membership function is placed in a new matrix called *classification matrix*. It is a matrix that has the same size as the image and the same dimensions. The values of the matrix elements represent the classes of the pixels of the corresponding image.

The EM algorithm is used in different image segmentation problems, such as medical images, natural scene images, and texture images. The authors in [CD00] presented an enhancement segmentation of texture images by the EM algorithm. The basic idea behind their algorithm is to minimise the expected value of the number of misclassified pixels by EM estimates using the Maximization of the Posterior Marginals (MPM) of the classification. After each iteration of the EM algorithm the MPM uses the estimated parameters to maximise the conditional probability of the classification of a certain pixel given its observed value.

The authors in [LFK08] proposed image segmentation for computed tomography (CT) images in two phases. First, a contrast enhancement is applied to cut off the skull and the background from the image. The EM is then used to extract the brain matter and the cerebrospinal fluid CSF.

In [MNG07] the EM algorithm is used to segment radiogram images taken of welded joints. The classes are mainly the weld defect, the welded joint, and the base metal. The results obtained are stable and satisfactory.

The author in [Yam98] introduced the use of the EM in colour image segmentation. The segmentation process was carried out for each colour subband of the RGB colour space. A similar recent work is introduced by the authors in [HL07], where the used colour space is the HSV.

The EM was also combined with other methods such as the continuous edge process in [KWSW97] for MRI segmentation.

Fuzzy-based Methods

The segmentation problem can also be reformulated as a classification problem. The isolation and extraction of the objects from a complex scene is done by classifying them into groups.

Fuzzy sets are a generalisation of conventional sets, which contain objects which have precise properties required for membership. For a conventional set H the membership function m_h tells either if a certain object belongs to the set or not. However, for a fuzzy set F the membership function m_f measures degrees to which objects satisfy imprecisely defined properties.

Farag et al. [MAF02] proposed a fully automatic technique to obtain image clusters. A modified fuzzy *C-means* (FCM) classification algorithm is used to provide a fuzzy partitioning. The FCM [BC77] for image segmentation has several advantages: *i*) it is unsupervised, *ii*) it can be used for an arbitrary number of features and classes, *iii*) it distributes the membership function in a normalised fashion and *iv*) it utilises the fuzzy sets to model the uncertainty in class definitions while clustering. The authors proposed a modification by introducing $R_{i,k}$ as the resistance of pixel x_i to be clustered to class k. This resistance can be tolerated by the neighbouring pixels x_j, where $j \in neighbourhood(i)$. The neighbouring pixels work to decrease the pixels resistance value by a fraction $u_{j,k}$ that depends on the membership value of x_j to the class k. As the system converges, its minimum membership value reaches its meaningful value, and the neighbouring pixels effect robustly the result. The total effect of the neighbouring pixels is that each surrounding pixel tries to pull its neighbour towards its class without neglecting the effect of the pixel itself. Farag et al. applied the modified FCM algorithm on medical data such as CT brain scans or a single channel MRI. The results show superiority over that of the FCM algorithm even with the pre-processing step.

A general disadvantage of the FCM is its sensitivity to the number of features used for the similarity function. The smaller the number of features for the similarity function the more acute are the class centres. For a high number of features it is possible that every object is to be classified to a separate class which means that the results may become blurred. Another disadvantage is its computational complexity. Hence, in the application of the algorithm the number of objects as well as the number of object's features must not be too large [Poh04].

In recent years some authors have also used the idea of image fuzziness to develop new thresholding techniques [PR88, Tiz98]. For example, a membership function m_f is moved value by value over the existing range of grey levels. In each position, a measure of fuzziness is calculated. The position with a minimum amount of fuzziness can be regarded as a suitable threshold.

In a recent work the use of the k-means algorithm in simple image segmentation procedures for general-purpose images is proposed [CW04]. First, the image is partitioned into non-overlapping blocks of size 4×4 pixels. A feature vector is then extracted for each block representing colour and spatial variations. Each feature vector consists of six features. Three of them are the average colour components in the block from the LUV colour space. The other three represent the square root of the energy of the horizontal, vertical and diagonal details obtained by a one-level wavelet transform. The k-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one region in the segmented image. The algorithm does not need to have the number of clusters K a priori. It selects K adaptively by gradually increasing K until a stopping criterion is met. The number of clusters in an image changes in accordance with the adjustment of the stopping criteria.

As a drawback no information about the spatial layout of the image is used in defining the regions, so they are not necessarily spatially contiguous. Because the focus of this work was not to achieve superior segmentation results but good categorisation performance, more attention is given to propose an image segmentation procedure with low computational costs.

Neural Networks-based Methods

For any artificial vision application, one desires to achieve robustness of the system with respect to random noise and failure of processors. Moreover, a system can be made artificially intelligent if it is able to emulate some aspects of the human information processing system. Another important requirement is to have the output in real time. Methods based on neural networks are attempting to achieve these goals.

Neural networks are massively connected networks of elementary processors. This architecture usually makes the system robust while the parallel processing enables the system to produce output in real time.

Neural networks have been used in different ways in the area of image segmentation. For example, a multilayer neural network has been used to segment noisy images. The output status of the neurons in the output layer has been viewed as a fuzzy set. The weight updating roles have been derived to minimise the fuzziness in the system. In this way, this algorithm integrates the advantages of both fuzzy sets (decision from imprecise and incomplete knowledge) and neural networks (robustness) [DB00].

The purpose of the study in [AJN97] is to explore the potential of Learning Vector Quantization (LVQ) in Artificial Neural Networks (ANN) for the classification and segmentation of magnetic resonance (MR) images of the brain. Learning Vector Quantization in ANN is a classification network that consists of two layers. The two-layer ANN classifies patterns by using an optimal set of reference vectors or *code words*. A code word is a set of connection weights from input to output nodes. Its label defines the classification of the input vector. In the training phase the labels of the code words are determined by presenting a number of input vectors with known classification and assigning the code words to different classes by majority voting. The authors used multi-spectral MR images and demonstrated an improved differentiation between normal tissue types of the brain and surrounding structures using pixel intensity values and spatial information of neighbouring pixels.

In [GYB02] the authors proposed a tool for segmenting scanned colour artwork images. They proposed a neural network to categorise pixels into specific groups using the surrounding context in which the pixels are located. The neural network operates on a context of colour pixels and decides how a small group of pixels at the centre of this context should be divided (if at all). The authors considered five factors to achieve an efficient solution: *i*) minimising the number of inputs required by the neural network, *ii*) establishing a concise output format, *iii*) optimising the structure and training ability of the neural network, and two more factors regarding the implementation of the network.

Considering the first factor. The more inputs required by a neural network, the larger it becomes and the more computational effort is necessary. Thus it is desirable to implement neural networks with as few inputs as possible while still allowing it to produce favourable results. Here each pixel from the RGB colour space and a context around the pixel of size 8×8 and 16×16 pixels were considered in the processing to segment each pixel. The average colour (for each R, G, and B component) for the entire context (e.g., a 16×16 pixel window) is computed. Then the relative difference from this average is computed for each pixel within the context.

This provides signed colour difference values for each colour component ranging from -255 to +255. This is used as a scheme to maintain the necessary semantics or relationships between pixels within the context. It is also consistent with human vision in general since image information is always provided relative to a surrounding context.

Considering the second factor. The smaller the number of unique responses a neural network is required to provide, the more practical and trainable it becomes. For reasons similar to those stated in 'first factor', it is also useful to limit the output of a neural network using as few "bits" as possible. The neural network here is intended to provide detailed boundary information. When these edges form continuous, closed contours the areas bounded within are considered as segments.

Considering the third factor. A constructive algorithm driven by genetic algorithms is used to define the topology of a neural net.

Generally, the size and computational requirements of a neural network can quickly grow without bound making the approaches impractical for many types of image processing tasks [GYB02].

Support Vector Machine-based Methods

Support vector machines (SVM) are one of the most recent algorithms in artificial neural networks. This new learning algorithm was proposed by Vapnik and is based on the statistical learning theory [Vap98]. "Most classical neural network algorithms require an *ad-hoc* choice of system's generalization ability, the SVM approach proposes a learning algorithm to control the generalization ability of the system automatically". SVM have been used for pattern recognition, regression estimation, density estimation, and ANOVA decomposition [Bur98].

SVM are successfully used as a technique for data classification. A classification task usually involves training and testing data which consist of some data instances. Each instance in the training set contains one *target value* (class label) and several *attributes* (features). The goal of SVM are to produce a model which predicts target values of data instances in the testing set which are given only the attributes. The training data is a set of instance label pairs (x_i, y_i), $i = 1, ..., l$ where $x_i \in R^n$ and $y \in \{1, -1\}^l$ in case of two-classes classification. The training vectors x_i are mapped into a higher feature space through some non-linear function say ϕ. In this space the SVM construct an optimal linear separating hyperplane with the maximal margin [HCL03]. This decision hyperplane is determined by certain points of the training set, termed *Support Vectors* [Bur98].

In [ZHT05] the authors propose an image segmentation method based on SVM in indoor environments. The purpose is to segment the images of seat numbers taken by a mobile robot in order to know where it is. The mobile robot has a CCD camera which captures RGB colour images of size 240×360 pixels. It is used to measure the relative location between the floor landmarks which in this case are the seat numbers of the desk. The images are considered to be of two classes. One is the number region, i.e., the object, and the other is the remaining region, i.e., the background. The inputs of the SVM are x^1, x^2, and x^3, which are the red, green, and blue components of each pixel of the image. In order to speed up the training of the SVM the size of the images was reduced to 1/9 of the original size, i.e., 8×120 pixels. An image was used to train the networks making the number of training sets 960 pixel points. The introduced experiments show very good classification of 99.91% in the training phase with 59 support vectors when the radial basis function (RBF) was used as a kernel.

The support vector machines share the high computational requirements with other types of neural networks. Moreover, they need a training data set, which may not be available in all applications.

2.2.3 Region-based Approaches

The methods belonging to region-based approaches use the characteristic of closed regions in the pixel-class mapping. All the pixels that are classified to an object or a region must specify certain homogeneity conditions as well as have spatial connectivity. The pixels from separate parts of the image are assigned to different objects or regions even if they meet the same homogeneity conditions.

Edge Detection

The segmentation based on edge detection classifies the pixels laying inside a region closed by edges to a class different to those laying on the other side of the edges. The edge detection segmentation methods recognise not the homogeneous parts of the image but the inhomogeneity between the parts.

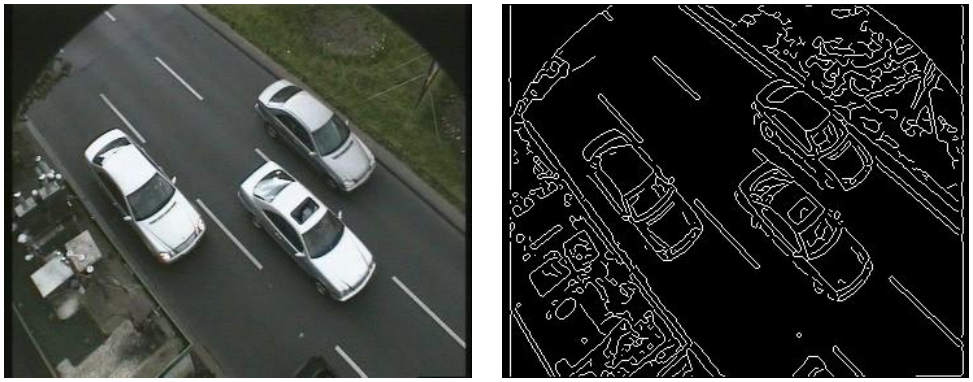
The basis of the edge detection operation is the use of a gradient operator that determines the level of variance between different pixels. The gradient is the change in grey level with direction.

Vertical edges would be detected by calculating the horizontal gradient and horizontal edges by calculating the vertical gradient. Diagonal edges response partially to both the horizontal and vertical edge detectors. The created image could be called a *gradient image*. Edges can be detected by using a gradient operator followed by a threshold operation to detect the extreme values of the gradient [Rho07]. Examples of gradient operators are Roberts, Prewitt, and Sobel operators.

The method of Canny [Can86] optimises the edge detection by defining three objectives for the detectors: First, finding "true" edges between objects and avoiding the weak ones that may appear due to noise. Second, minimising the distance between the estimated edges and the real edges. Third, joining the disconnected edges to be represented only by one edge string [Can86, Poh04].

To achieve these objectives the detector uses an iterative algorithm that starts with smoothing by the Gaussian filter. The purpose of the smoothing is to suppress the weak edges. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. Then the gradient value and direction are estimated for each pixel. To have a thin line as edges, the edges found are traced along the edge direction afterwards and the non-maximum suppression is used to suppress any pixel value that is not considered to be an edge. Finally, a hysteresis thresholding procedure is done to eliminate the discontinuity of the edges. Two thresholds are used. All the edge pixels that are greater than the higher threshold are counted as strong edges. The edge pixels under the lower threshold are counted as noise and are eliminated from the resulting image. The edge pixels that are connected to strong edges and greater than the lower threshold are counted as weak edges but considered in the resulting image. Fig. 2.5(b) shows the output using Canny detector for the image shown in Fig. 2.5(a).

In [YWC05] an improved gradient threshold edge detector is introduced. The algorithm smooths and differentiates the image to get the gradient image. Then the edges are labelled with a low threshold to get the prime edge image, which includes all the potential edges. To get the perceptive edges, the gradient image is further masked with the local luminance and activity, and then edge labelling is performed on the masked gradient image using another threshold. The keys of the improved method for detecting perceptive edges are to determine the mask regions and the local activities.



(a) (b)

Figure 2.5: Example of the Canny edge detector.

The authors in [LKK03] propose a new edge detection method using a 3×3 ideal binary pattern and lookup table (LUT) for the mobile robot localisation without any parameter adjustments. The mean of the pixels within the 3×3 block is used as a threshold by which the pixels are divided into two groups. The edge magnitude and orientation are calculated by taking the difference of the average intensities of the two groups and by searching the directional code in the LUT, respectively. Moreover, the input image is not only partitioned into multiple groups according to their intensity similarities by the histogram, but also the threshold of each group is determined by fuzzy reasoning automatically. Finally, the edges are determined through non-maximum suppression using edge confidence measure and edge linking.

An intelligent scissor method proposed by Mortensen and Barret [MB95] is a boundary detection method, which is an interactive contour detection method allowing the user to select interactively the most suitable boundary from a set of all optimal boundaries emanating from a seed point.

In [NH04] the authors present the implementation of an adaptive edge detection filter on a FPGA. The design is scalable to handle higher resolution images with a resolution of 256×256, while maintaining the clock frequency used to process standard video-resolution signals at 30 frames per second.

Region Growing

In this group the algorithms start with an interactive starting point. On the basis of this starting point the search of a coherent area takes place as a function of the defined neighbourhood and of a homogeneity condition which can be given by the user. In order to be able to have a successful segmentation by these procedures the area which is to be segmented must possess internally a larger homogeneity than at its borders. If several structures in an image are to be segmented, the starting point inquiry must be repeated several times. With the input of the starting points the user inserts his knowledge about the number of searched objects and their positions into the procedure. The definition of a homogeneity criterion can be more problematic for the user, he often formulates this knowledge vague and indistinct. Thus, it cannot be passed easily into a calculable form. Therefore, the definition of the homogeneity borders is sometimes a trial and error process.

The literature contains a large number of papers that describe how to find a suitable homogeneity criterion based on the associated applications. Most of the methods define the homogeneity criterion as a function of the grey level distribution in a region directly around the starting point. With this method the criterion is updated whenever a new point is added to the region [CL94]. Due to this, the set of the starting points and the selected search order are dependent. The grey level variability within the region to be segmented can cause an update of the criterion in such a way that after many processed points the criterion is no more fulfilled.

Another category of methods selects the homogeneity criterion on the basis of optimal consideration for complete segmenting of the entire image. Hence, for different regions of the image different homogeneity conditions are used [AB94, HK98]. With these procedures a complete segmentation must always be made, which leads to a higher computational complexity. On the other hand, this procedure has the advantage that no explicit homogeneity criterion has to be indicated, because this is defined by the algorithm. As by the proposed seeded region growing algorithm, the growth strategy is selected so that on the basis of several interactively entered starting points the regions are always increased directly around the regions neighbouring pixels, for which the difference between the mean grey level of the respective regions and that of the updated regions is smallest. The procedure is continued, until all pixels were assigned to a region. A drawback in this procedure is that the choice of the number of starting points affects completely the quality of segmentation of the individual regions.

Split and Merge

Split and merge is a development of region growing, in that the image is split and merged according to some homogeneity criterion which must be defined at the beginning of the process. The split and merge procedure represents a simple possibility for segmenting uncomplicated structures in images.

The procedure processes the image in the form of a quad-tree. First, the entire image is considered as one region. If this region does not fulfil the fixed homogeneity requirements, then it is divided into four subregions by moving down the tree one level. The procedure of the partitioning is recursively repeated as long as all determined regions fulfil the homogeneity criterion. The splitting terminates at the pixel level. If adjacent subregions have a similar homogeneity then they are merged by moving up one level. Thus, the output is an image that comprises blocks of varying size from different levels of this tree. The leafs of the tree represent homogeneous regions (see, e.g., [Toe05]).

A problem with this kind of segmenting is that the splitting using a relative homogeneity criterion is not always clear. Different results can develop according to whether the regions are merged first in horizontal or in vertical direction [Poh04].

2.2.4 Template-based Approaches

The idea of template matching is to create a model of an object of interest (the template, or kernel) and then to search over the image of interest for objects that match the template. For each pixel in the image, a similarity measure between the template and the underlying pixels in the image of interest is computed. A widely used similarity function is the normalised cross-correlation (NCC). A perfect match would give an NCC coefficient value of $+1$. Comparison with an exact negative (inverted) grey scale value gives an NCC coefficient of -1, while a comparison with a completely unrelated template gives an NCC coefficient of 0.

In [Ols02] a measure is introduced for template matching with a binary or grey scale template using a maximum likelihood. In this formulation, a function is generated that assigns a likelihood to each of the possible template positions.

For applications in which a single instance of the template appears in the image, such as tracking or stereo matching, the template position with the highest likelihood is accepted if the matching uncertainty is below a specified threshold. For other recognition applications, all template positions with likelihood greater than some threshold are accepted. The algorithm is used for object recognition, stereo matching, feature selection, and tracking.

The author in [Lun00] uses the entropy to enhance the performance of the similarity measurement while matching. Maximum entropy matching aims to increase the speed of the template matching while keeping the performance. It works by comparing derived image features and the template for each possible displacement of the template. To increase the speed of the matching algorithm the number of data which have to be compared should be as small as possible, but with as much information as possible. Therefore, the data used in the comparison should have high average information, that is, high entropy.

The authors in [TT07] present an approach to accelerate multi-scale template matching. The main computation saving is achieved by representing the template as a linear combination of a small number of Haar-like binary features. Each such feature has one rectangle which is defined by two of its corners. For an image of $W \times H$ pixels, the feature dictionary contains $H(H+1)W(W+1)/4$ such features. These bases functions have the advantage that the inner product of a data vector with a box feature can be performed by several integer additions, instead of N floating point multiplications, where N is the dimension of the feature. In addition, they have the advantage of ease for scale adaptation of the binary box features. Each such feature only needs to store two point locations corresponding to the left up and right down corner of the rectangle. As a result, the feature can be scaled to arbitrary size. The proposed representation is used to accelerate the normalised cross correlation algorithm.

2.2.5 Discussion

Pixel-based approaches based on statistical modelling are the most appropriate approaches if the image has many non-uniformly shapes and overlapping disconnected regions. In such a case other approaches, such as region-based or template-based ones, may not be suitable. Among the pixel-based segmentation algorithms the EM algorithm is a very powerful one.

However, the EM algorithm completely disregards information contained implicitly in the spatial coordinates of the pixels. Hence, it ignores the spatial correlation between the neighbouring pixels. Most probably it is an error if a pixel is estimated to be belonging to a class different than that of all neighbouring pixels. Therefore, a modification of the EM algorithm to consider the spatial correlation between the neighbouring pixels is a must. The multiresolution analysis can give an efficient pixel neighbouring dependency across both scale and space, and at the same time is computationally efficient.

The region growing procedure can be successful if the features of the regions to be segmented are very distinct from each other. The accuracy of the segmentation depends very strongly on the grey level contrast between the different regions.

In template matching approaches the segmentation process requires a predefined template which describes the objects of interest in the image. Due to the complex structure of some images, it is difficult to have a template before the segmentation. Moreover, template matching approaches are usually computationally expensive because the template needs to be matched to every location in the image and the matching involves element-by-element floating point multiplications, since most similarity functions involve the convolution of the template with the image.

The application investigated in the case of still image segmentation is the segmentation of the magnetic resonance image, MRI, of the human brain. The presence of impulsive and additive Gaussian noise, which is an inherent problem in the acquisition of MRI, the high intensity variation among pixels from the same class, and the partial voluming, which causes mixing of pure class intensity near the edges [AU96, MAF02], are all problems which cause the uncertainty to play a fundamental role in the potential image segmentation approach. Due to this uncertainty, an effective segmentation algorithm must address, in addition to the pixel intensity feature, the modelling of the image as a statistical field (Bayesian statistics) where the pixel intensity is known a priori, usually called the incomplete data, and the class of the pixel is the missing data.

For this purpose the use of the EM algorithm, which is well-known as an efficient tool to solve this type of problems, in combination with the multiresolution analysis is expected to give good results. This combination meets the constraints for the modelling of the MRI and overcomes the limitation of the EM.

2.3 Video Segmentation

2.3.1 Principles

Video segmentation is a necessary step for indexing videos based on content, object tracking, and monitoring. In most cases it is the aim to detect and extract objects of interest from video data. The video segmentation methods label pixels at each frame that are associated with independently moving parts of a scene. Different features and homogeneity criteria generally lead to different segmentations of the same data, for example colour, texture, or motion parameters. If the objective of the segmentation is to enable analysis of motion parameters this leads to what is called *motion segmentation*.

Motion segmentation is closely related to two other problems, *motion detection*, and *motion estimation*. Motion detection is a special case of motion segmentation with only two regions, namely changed and unchanged regions, if the scene is observed by a static camera, or global and local motion regions, if the scene is observed by a moving camera. In the last case, motion detection requires some sort of global or local motion estimation, either explicitly or implicitly [Bov00]. There are mainly three categories for video segmentation: temporal (frame) differencing, optical flow and background subtraction [ZZWF06]. Other methods can be grouped under other conventional methods.

2.3.2 Frame Differencing

The simplest method to detect change between two registered frames would be to analyse the frame difference (FD) image, which is given by:

$$FD_{t_c,t_r}(x,y) = s(x,y,t_c) - s(x,y,t_r) \tag{2.13}$$

where (x,y) denotes pixel location and $s(x,y,t)$ stands for the intensity value of pixel (x,y) in the frame at the moment t. The FD image shows the pixel-by-pixel difference between the frame at the current moment t_c and the reference image at the moment t_r.

The reference image s_{t_r} may be taken as the previous frame (successive frame difference), or an image at a fixed time, for example a background image that may be taken at a time when no moving objects were presented.

Assume for simplicity, that the illumination remains constant between the frames. The pixel location where $FD_{t_c,t_r}(x,y)$ differs from zero indicates "changed" regions as a result of local motion. In order to distinguish the non-zero differences that are due to noise from those that are due to local motion, segmentation can be achieved by thresholding the FD as:

$$MR_{t_c,t_r}(x,y) = \begin{cases} 1 & if \quad |FD_{t_c,t_r}(x,y)| > T \\ 0 & otherwise \end{cases} \tag{2.14}$$

where T is an appropriate threshold. MR is a labelled image, which is equal to 1 for changed regions and 0 otherwise. The value of the threshold T can be chosen by an optimal threshold determination algorithm. This pixel-wise thresholding is generally followed by one or more post processing steps to eliminate isolated labels. Post processing operations may include smoothing filters, discarding labels with less than a predetermined number of entries, and morphological filtering of the changed and unchanged region masks.

An important variation to frame difference is to add memory to the motion detection process. This can be achieved in a number of different ways, including temporal filtering (integration) of the intensity values across multiple frames before thresholding.

A frame difference with memory (FDM) can be formulated as:

$$FDM_{t_c}(x,y) = s(x,y,t_c) - \bar{s}(x,y,t_c) \tag{2.15}$$

where $\bar{s}(x,y,t_c) = (1-\alpha)s(x,y,t_c)+\alpha\bar{s}(x,y,t_c-1), t_c = 1,2,...,$ and $\bar{s}(x,y,0) = s(x,y,0)$. The value α, $0 < \alpha < 1$, is a constant. After processing a few frames, the unchanged regions in $\bar{s}(x,y,t_c)$, maintain their sharpness with a reduced level of noise, while the changed regions are blurred [Bov00].

A general problem with frame differencing is that its outcome usually contains both the moving object's contours and the unwanted contours of the background texture.

In [ZZWF06] the authors propose a three frame-differencing method to identify the contours of the moving objects for traffic monitoring applications. They assume that the contours of the background texture can be eliminated and the contours of the moving objects can be raised by finding out the contours whose counterparts in the two relevant two frame-differenced images coincide with each other. The operation is detailed in the following equation:

$$D(x, y, \Delta t) = |f(x, y, t) - f(x, y, t-1)| \cdot |f(x, y, t) - f(x, y, t+1)| \quad (2.16)$$

where $f(x, y, t-1)$, $f(x, y, t)$, and $f(x, y, t+1)$ denote a pixel of the preceding frame, the current frame and the sequent frame of a gradient-based image sequence, respectively.

2.3.3 Optical Flow

Temporal segmentation based on optical flow uses motion information deduced from consecutive frames. Optical flow means the flow of the displacement of the grey level intensities in an image sequence, which arises from the relative motion of objects and the viewer. Since the intensities in the image are the results of reflections on objects, the estimated optical flow can be grouped into regions or blocks that correspond to different objects. The estimation of the optical flow is based on the *grey value constancy assumption*: the grey value of a pixel is not changed by the displacement. In other words, if the brightness at the point (x, y) in the image plane at time t denoted by $I(x, y, t)$ remains constant, then at a point a small distance dx and dy away, and a small time dt later holds:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (2.17)$$

Expanding the left hand side by a Taylor expansion:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \ldots \quad (2.18)$$

yields:

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \ldots = 0 \quad (2.19)$$

Let $u = dx/dt$ and $v = dy/dt$ represent the speed of the moving object in the x and y directions and let dt tends to zero, then:

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad (2.20)$$

$$I_x u + I_y v + I_t = 0 \quad (2.21)$$

$$I_x u + I_y v = -I_t \quad (2.22)$$

Eq. 2.22 is called the *optical flow constraint equation* [HS92] and can be rewritten as:

$$(I_x, I_y) \cdot (u, v) = -I_t \tag{2.23}$$

Thus, the component of the image velocity in the direction of the image intensity gradient is:

$$(u, v) = \frac{-I_t}{\sqrt{I_x^2 + I_y^2}} \tag{2.24}$$

A problem of this calculation is that it is not possible to estimate the optical flow at right angles to this direction. This ambiguity is known as the *aperture problem*.

A classical approach first consists in estimating a dense motion field and then partition the scene only based on the obtained motion information, where the adjacent video components corresponding to the motion vectors are merged to form the meaningful video objects if they obey the same Hough or affine transformation motion model [WA94]. However, dense field motion vectors are not very reliable for noisy data [DM01, FZBH05]. In order to avoid the noise of optical flow, some techniques first include a change detector. However, the change detector introduces holes in uniform regions [FZBH05].

The authors in [KDM06] used an accumulation process of optical flow vectors computed by the Lucas-Kanade tracker [LK81] to detect the active traffic area and to predict the driving directions which constrain object movements. To calculate the optical flow between successive video frames the well-known combination of feature selection as introduced by Shi and Tomasi [ST94] and the algorithm of Lucas and Kanade for feature tracking [LK81] is used. Feature selection finds image blocks which are believed to allow the accurate estimation of the optical flow translation vector. The Shi-Tomasi algorithm utilises the smallest eigenvalue of an image block as criterion to ensure the selection of features which can be tracked reliably by the Lucas-Kanade tracking algorithm. Finally, the results are filtered for relevance and quality. To reduce the number of mismatched motion vectors the root mean square error (RMSE) of each pair of blocks is evaluated and vectors with high error values are discarded. Afterwards, an accumulative process is performed on the resulted optical flow vectors. The extracted vectors at each pixel are collected and averaged over time. This way it is possible to retrieve a time averaged optical flow field consisting of an average motion vector for each image pixel.

A histogram based approach is used to allow more than one typical direction for each image pixel. The histogram is grouped in 12 bins, each covering an angle of 30 degree. By this histogram the dense information about the motion directions is obtained. In Fig. 2.6 the optical flow is coded by different grey levels to illustrate the average angles of the motion vectors.

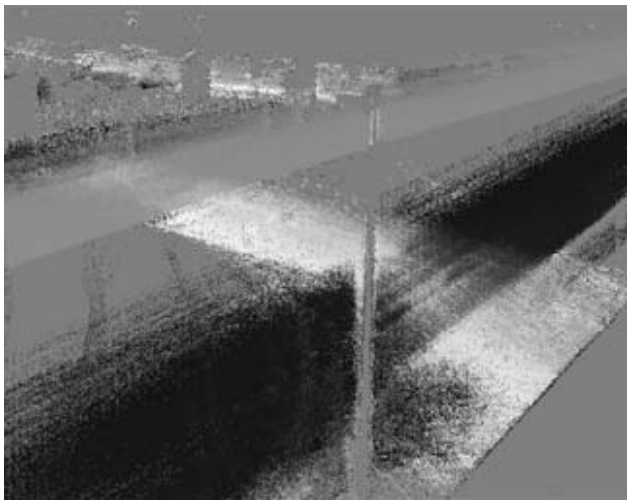
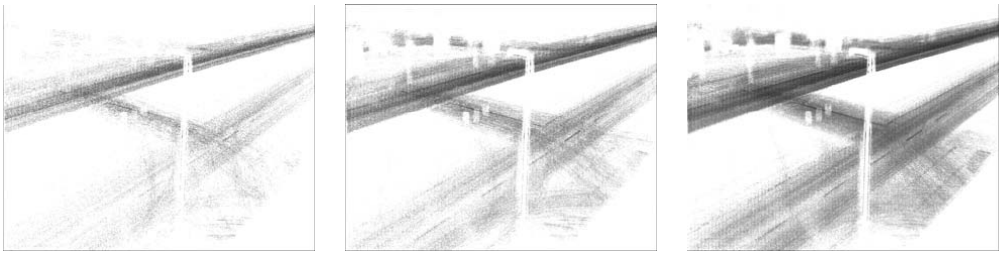


Figure 2.6: Optical flow. Different grey levels indicate different motion directions in Rudower Chaussee [KDM06].

A sufficient observation time is required to produce a dense and reliable averaged optical flow field. In general, the field quality will increase as long as the traffic situation in the observed scene remains constant. First experiments indicate that the processing of at least 50,000 frames is necessary. Fig. 2.7 illustrates the number of motion vectors which have been obtained at each location.

2.3.4 Background Estimation and Subtraction

Many papers dealing with moving object detection in traffic surveillance applications are based on background subtraction [CLK+00, YYK03, ZK03a, ZK03b, BBRS04, TCAA05]. Typically, a background image is created and updated by the current frame of the image sequence. Then a mask is created from the difference between the current traffic scene image and the updated background image. This mask is used to extract the region of interest which represents the moving traffic objects.



(a) (b) (c)

Figure 2.7: Frequency of movement per pixel after (a) 10000, (b) 25000, and (c) 50000 frames [KDM06].

The background can be updated recursively from the actual image data as in [CLK+00, TCAA05]. The algorithm classifies the pixels of a current frame into either pixels belonging to a moving object or pixels belonging to the background. The grey values of the background pixels are used to update the current estimation of the background. For all other pixels the values from the previous estimation are used as they are. Let $I_t(x, y)$ represent the intensity value at pixel position (x, y) at the time t in the image frame I_t. The new estimation of the background intensity value at the same pixel position $B_{t+1}(x, y)$ is calculated as follows:

$$B_{t+1}(x,y) = \begin{cases} \alpha B_t(x,y) + (1-\alpha) I_t(x,y) \\ \quad \text{if the pixel at } (x,y) \text{ is non-moving} \\ B_t(x,y) \\ \quad \text{if the pixel at } (x,y) \text{ is moving} \end{cases} \tag{2.25}$$

where α is an update parameter that specifies how fast new information supplants old one. The initial estimate of the background $B_0(x, y)$ is set as the first image frame $I_0(x, y)$. Then the parameter α is set as a positive real number close to one.

Assigning a pixel to a moving or non-moving region is done with the help of a threshold. It is assumed that a pixel belongs to the moving regions if it differs significantly from the estimated background at the same time instant:

$$(x,y) \in \text{Moving region} \quad if \quad |I_t(x,y) - B_t(x,y)| > T_t(x,y) \tag{2.26}$$

where $T_t(x, y)$ is the threshold at the time t.

The threshold is statistically determined based on the scene and considered in different work as an application-dependent parameter such as in [YYK03, ZK03a, ZK03b, BBRS04]. It can also be estimated recursively in a similar manner as the background as follows [CLK+00, TCAA05]:

$$T_{t+1}(x,y) = \begin{cases} \alpha B_t(x,y) + (1-\alpha)(\beta |I_t(x,y) - B_t(x,y)|) \\ \quad \text{if the pixel at } (x,y) \text{ is non-moving} \\ T_t(x,y) \\ \quad \text{if the pixel at } (x,y) \text{ is moving} \end{cases} \tag{2.27}$$

where β is a real number greater than one and the update parameter α is a positive number close to one. Initial threshold values are set to an experimentally determined value. As it can be seen from the last equation, the higher the parameter β, the higher the threshold or the lower the sensitivity of the detection scheme.

In [CLK+00] the value of β is set to a constant value 5. That means that $T_t(x,y)$ is analogous to 5 times the local temporal standard deviation of intensity of the pixel at (x,y). A three-frame differencing operation is performed to determine regions of legitimate motion, followed by adaptive background subtraction to extract the entire moving region. The three-frame differencing role suggests that a pixel is legitimately moving if its intensity has changed significantly between both the current image and the last frame, and the current image and the next-to-last frame:

$$\begin{aligned} &|I_t(x,y) - I_{t-1}(x,y)| > T_t(x,y) \text{ and} \\ &|I_t(x,y) - I_{t-2}(x,y)| > T_t(x,y) \end{aligned} \tag{2.28}$$

Then the blob b_t can be filled by taking all the pixels in R that are significantly different from the background model B_t:

$$\begin{aligned} b_t = \{&(x,y) : |I_t(x,y) - B_t(x,y)| > T_t(x,y) \\ &(x,y) \in \text{bounding box of } R_t\} \end{aligned} \tag{2.29}$$

where

$$\begin{aligned} R_t = \{&(x,y) : (|I_t(x,y) - I_{t-1}(x,y)| > T_t(x,y)) \\ &(|I_t(x,y) - I_{t-2}(x,y)| > T_t(x,y))\} \end{aligned} \tag{2.30}$$

Background Modelling Using Gaussian Mixture Model

The Gaussian mixture model belongs to a class of density models which have several functions as additive components. These functions are combined together to provide a multimodal density function, which can be employed to model intensities of a dynamic scene or object.

The authors in [SG99] proposed to use a mixture of Gaussians distributions to model the background for the tracker module of a video surveillance system. This technique models each background pixel as a mixture of K Gaussian models. The Gaussians distributions are evaluated using a simple heuristics to hypothesise which are most likely part of the background process. The probability of observing the current pixel value is:

$$P(X_t) = \sum_{i=1}^{K} w_{i,t}\eta(X_t; \mu_{i,t}, \Sigma_{i,t}) \tag{2.31}$$

where

K is the number of distributions, usually between 3 to 5 in practice
$w_{i,t}$ is an estimate of the weight of the i^{th} Gaussian in the mixture at t
$\mu_{i,t}$ is the mean value of the i^{th} Gaussian in the mixture at time t
$\Sigma_{i,t}$ is the covariance matrix of the i^{th} Gaussian in the mixture at time t
η is a Gaussian probability density function.

Every new pixel value X_t is checked against the existing K Gaussian distributions until a match is found. Based on the matching results the background is updated as follows: If X_t matches component i, that is X_t is within the range of ± 2.5 standard deviations from the mean of the distribution, then the parameters of the i^{th} component are updated as follows:

$$w_{i,t} = w_{i,t-1} \tag{2.32}$$
$$\mu_{i,t} = (1-\rho)\mu_{i,t-1} + \rho I_t \tag{2.33}$$
$$\sigma^2_{i,t} = (1-\rho)\sigma^2_{i,t-1} + \rho(I_t - \mu_{i,t})^T(I_t - \mu_{i,t}) \tag{2.34}$$

where $\rho = \alpha\eta(X_t|\mu_{i,t-1},\ \Sigma_{i,t-1})$.

The parameters for unmatched distributions remain unchanged:

$$w_{i,t} = (1-\alpha)w_{i,t-1} \tag{2.35}$$
$$\mu_{i,t} = \mu_{i,t-1} \tag{2.36}$$
$$\sigma^2_{i,t} = \sigma^2_{i,t-1}. \tag{2.37}$$

If X_t matches none of the K distributions, then the least probable distribution is replaced by a distribution where the current value acts as its mean value. The variance is chosen to be high and the a-priori weight is low [ZK03a]. The background estimation problem is solved by specifying the Gaussian distributions, which have the most supporting evidence and the least variance. In order to represent background processes, first the Gaussians are ordered by the value of $w_{i,t}/\left\|\Sigma_{i,t}\right\|$ in decreasing sequence, because the pixels belonging to the moving objects have larger variance than a background pixel. The background distribution is on top with the lowest variance by applying a threshold T. All pixels X_t which do not match any of these components will be marked as foreground, i.e., belonging to moving region.

The authors in [AZ08] propose an enhancement to avoid missegmentation when the object is in contact with parts of the background having the same appearance as the object. In this case, the segmentation based on the region information alone can result in contour being distracted and deviating from the true object boundaries. So, [AZ08] propose to add boundary and shape information about the desired objects to the segmentation model. Indeed, there is generally a strong correlation between the object boundaries and the image edge map. This feature is used to constrain the alignment of the object contour with strong image edges. On the other hand, the object shape does not generally change abruptly between successive frames of the sequence. Thus, adding a shape constraint that prevents large changes in the object shape can enhance the robustness of segmentation against distraction.

Background Estimation Using 2D Wavelet

The method proposed by Töreyin et al. [TCAA05] is based on a background updating algorithm and the 2D wavelet analysis to extract a moving traffic object as a region of interest (ROI). Therefore, a mask is created from the difference between the current traffic scene image and the estimated background image. This mask is used to extract the ROI. The background is estimated recursively as proposed in [CLK$^+$00]. The estimation is considered to be recursive because the threshold for distinguishing background pixels and moving object pixels is updated recursively. However, instead of estimating the background based on the image sequence, the authors proposed an estimation based on the wavelet transform coefficients at the third level. An updated background image and a mask are computed for each wavelet subband and for each frame.

The estimated background D_{t+1} of the subband image $J_t(x, y)$ at time instant $t + 1$ is computed similar to Eq. 2.25 as follows:

$$D_{t+1}(x,y) = \begin{cases} \alpha D_t(x,y) + (1-\alpha) J_t(x,y) \\ \quad \text{if the pixel at } (x,y) \text{ is non-moving} \\ D_t(x,y) \\ \quad \text{otherwise} \end{cases} \tag{2.38}$$

where $J_t(x, y)$ is a subband image resulting from the wavelet analysis of the image frame at time t and D_t is the associated background image. The associated initial background estimation D_0 is assigned to be the subband image of the first frame of the video. The parameter α has an effect on the stability of the estimation of the background and is global for all subbands. Its value is a trade-off between accelerating the update of the background and the sensitivity to slow motions and stopped objects. For each subband image $J_t(x, y)$ there is an associated threshold $T_t(x, y)$, which is recursively updated as in Eq. 2.27. This threshold is then used to classify the pixels into moving or non-moving pixels.

All wavelet coefficients satisfying the inequality $|J_t(i, j) - D_t(i, j)| < T_t(i, j)$ are classified as belonging to a moving object. The extracted moving objects from the different subbands can then be fused to a ROI within the current frame. The last step of the algorithm is a simple region growing to include classified pixels. All steps of the algorithm are shown in Fig. 2.8.

2.3.5 Other Conventional Methods

In addition to the methods that are based on temporal change for video segmentation, some conventional image segmentation techniques are used.

Thresholding-based Methods

Other approaches use the thresholding technique which is based on the notion that vehicles are compact objects having different intensities as their background, e.g., [Par01]. A general problem in this approach is that it would not avoid false detection of shadows or missed detection of vehicle parts with similar intensities as its environment. Binary and grey-scale morphological operators were found to improve background-foreground segmentation results as in [WNL01].

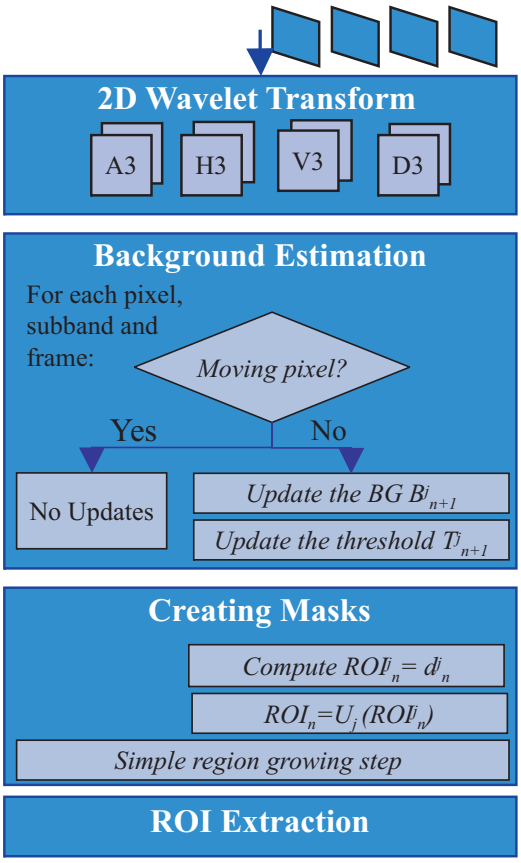


Figure 2.8: Block diagram of the 2D wavelet-based algorithm for video segmentation corresponding to the method of Töreyin et al. [TCAA05].

Edge Detection-based Methods

A new approach to compute consistency of edge detection on video segmentation is introduced in [ACHTSN06]. The approach is based on two main changes to original Canny's proposal: the substitution of the Gaussian filter step with an anisotropic diffusion filter [PM87] and the replacement of the hysteresis step with a dynamic thresholding operation. The use of the Canny detector, which has been mainly designed for still images, introduces instability situations in detected edges during the sequence. These can be noticed as blinking edges, i.e., edge features appear in one frame but suddenly disappear in the next one. Some other methods are based on edge detection as a first step followed by fitting a proposed model as in [KM03]. If there are objects that are not considered in the model and took part in the video data, they cannot be detected.

Region Growing-based Methods

Region growing methods are also used for video segmentation. Similar to the case of still images, the first step is to select seeds, which often determines the final segmentation results by subsequent region growing. It is expected to have a human interaction for selecting the initial seeds. This is a major drawback for video segmentation since in case of changing the scene, a new human interaction will be required.

In [FFJ05] the authors propose an automatic region growing algorithm for semantic video object segmentation for content-based multimedia applications. They use a competitive learning neural network to do the initial segmentation. Each frame in the video is divided into blocks, each of which contains 8×8 pixels. For each block an average value is computed and called *DC* coefficient. The coefficients are then fed to a competitive learning neural network to decide if the corresponding block is an object block or a background block. The first five video frames are used as training sequence. The resulting classification is used as a guide for initial seed selection. Afterwards, a skeleton for the segmentation is generated. The non-reliable pixels on the boundaries of the object region are removed, and those pixels located in the middle of the object region between two boundaries are kept. As a result, the process generates a thin skeleton for each object region. Similarly, the process is repeated for background regions. Correspondingly, the pixels on the object skeleton are selected as the seeds for object region growing, and the pixels on the background skeleton are selected as the seeds for background region growing. The rest of the work is done as in [AB94].

2.3.6 Discussion

Frame differencing is a simple direct method to detect motion as a change between successive frames from video data. However, it suffers from the extraction of unwanted regions from the background. This is due to the change in illumination and the moving of non-interesting objects in the background such as trees, clouds and so on.

The methods based on the optical flow give not only information about the sizes and locations of the moving objects in the scene, but also information about the direction and velocity. The movements that take place in the background or by objects that are not interesting for the monitoring process can easily neglected by the use of a simple velocity threshold.

However, these methods face two main problems in addition to the computation complexity. First, if the images are in low overall contrast because of a bad ambient conditions, then the number of significant vectors will decrease, which inhibits the grouping of vectors in blocks and hence inhibits a meaningful segmentation result. Second, a long observation time is required to produce a reliable optical flow field for the active traffic area.

The methods based on background estimation and subtraction are dependent on the update parameters and the initialisation of the background. Small values of the updating parameters imply an integration of the moving objects that are not active for a while as a part of the background. On the other hand, if the values of the updating parameters are too high, the methods need a relative long interval for a stable estimation of the background and they fail to adapt to the changes in the illumination. The update parameters and the initial background need to be set before the running of the methods by an expert. They have to be set carefully and for every new scene.

The application investigated in the case of video segmentation is moving object detection for traffic monitoring using a stationary camera. For this purpose, a method that is based on the frame differencing may lead to robust results if not only the temporal changes are considered but also the spatial changes, and if the active area from the scene are processed differentially from the background. The method must be computationally efficient and must offer segmentation with simple operations that can be easily implemented in hardware.

The use of the 3D wavelet transform meets these needs. It is able to analyse the input data spatially as well as temporally. Moreover it is hardware friendly, since it can be implemented by simple arithmetic operations.

Chapter 3

The Multiresolution Image Analysis

The subject of multiresolution analysis is to be found usually as a part of the fundamentals of the wavelet analysis. However, the term *Multiresolution* is much older than the wavelet transform. Some references go back to the end of 1970s. One of the first workshops titled "Multiresolution Image Processing and Analysis" was held in Leesburg, VA, USA on July 1982 [Ros84b].

In this chapter the multiresolution analysis is introduced as an independent concept instead of being a part of the wavelets. Starting with the formal definition, examples from early and recent work are given.

3.1 Introduction

Multiresolution analysis of a signal is a successive coarser approximation of the original signal. This can be interpreted as representing the signal by different levels of resolution. Each level contains information about different features of the signal. Finer resolution shows more details, while coarser resolution shows the approximation of the signal and only strong features can be detected.

A function or signal can often be viewed as a composition of a smooth background and actions or details in the foreground. The distinction between a smooth part and details is determined by the resolution. At a given resolution a signal is approximated by ignoring all details with higher resolution. Consider progressively increasing resolution, at each stage of the increase in resolution finer details are added to the coarser description providing a successive better approximation to the signal.

Eventually, when the resolution goes to infinity, the exact signal is recovered. Vice versa, for progressive decrease in the resolution at each stage more details are lost from the signal. This provides a successive coarser approximation of the signal until a point is reached where only one value describes the whole signal: the global average. Fig. 3.1 shows an image of a traffic scene in its original sampling resolution and two approximations.

(a) (b) (c)

Figure 3.1: Image in multiresolution representation. (a) Original resolution. (b) Approximation in one lower resolution level. (c) Approximation in three lower resolution levels.

The most obvious advantage of multiresolution representations is that they provide a possibility for reducing the computational cost of various image processing operations. For example, they can be used to perform coarse feature detection operations, such as spot or bar detectors, by applying the corresponding fine feature detection operations at a higher level [Ros84a].

The information contained in a signal is distributed into the levels of the resolution. Local information may be better processed in the high resolution levels, while global information may be processed in the low resolution levels. Working in a cross-resolution manner can help to process each type of information of the signal.

3.2 Multiresolution Representation

In this section we introduce the mathematical background of the multiresolution analysis and follow [LOPR97, Bla98, Röm07].

A multiresolution analysis consists of a sequence of successive approximation spaces. More precisely, consider a set of closed subspaces $V_j,\ j \in \mathbb{Z}$ from $L^2(\mathbb{R})$ satisfy the following conditions:

1. A smaller value of j means finer resolution. An approximation A_j in a finer resolution contains all information of any lower level approximation A_i, where $i > j$. Then a subspace V_j is contained in all the lower subspaces:
$$V_{j+1} \subset V_j \subset V_{j-1} \subset ... \subset L^2(\mathbb{R}), \forall j \in \mathbb{Z} \tag{3.1}$$

2. As the resolution increases, more details are included and the original signal is at least obtained:
$$cl(\bigcup_{j\in\mathbb{Z}} V_j) = L^2(\mathbb{R}) \tag{3.2}$$

3. As the resolution decreases, more details are removed and the information contained goes to zero:
$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\} \tag{3.3}$$

4. The approximations are scaled versions of each other:
$$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j-1}, \forall j \in \mathbb{Z}$$
i.e.,
$$f(t) \in V_j \Leftrightarrow f(2^j t) \in V_0, \forall j \in \mathbb{Z} \tag{3.4}$$
This condition is the key requirement of scale invariance.

The vector space V_j can be interpreted as the set of all possible approximations at a resolution j of the functions in $L^2(\mathbb{R})$, the space of square-integrable functions. Thus, in V_0 are all functions without any approximations.

5. The subspace V_0 is invariant regarding integer translations, i.e.,
$$f(t) \in V_0 \rightarrow f(t-k) \in V_0, \forall k \in \mathbb{Z} \tag{3.5}$$

Any set of vector spaces $V_j,\ j \in \mathbb{Z}$ which satisfies the properties of Eqs. 3.1 - 3.5 is called a multiresolution approximation of $L^2(\mathbb{R})$. An approximation A_j of any signal $f(t)$ at a resolution j must be associated with this set of vector spaces [Mal89].

To compute the approximation A_j of the signal $f(t)$ at resolution j, an orthonormal basis is needed that projects the signal in V_j. Let the set $\{2^{-j/2}\phi_{j,k}(t), k \in \mathbb{Z}\}$ be an orthonormal basis of $V_j, j \in \mathbb{Z}$ where:

$$\phi_{j,k}(t) = \phi(2^{-j}t - k) \tag{3.6}$$

The function $\phi(t) \in L^2(\mathbb{R})$ is called the scaling function of the multiresolution analysis.

Consider the properties in Eqs. 3.2 and 3.3. Computing an approximation of $f(t)$ at level j, some information about $f(t)$ is lost. All details on scales smaller than 2^{-j} are suppressed. However, as j decreases to zero the resolution increases and the approximated signal should converge to the original signal. Conversely, as the level increases to $+\infty$, the approximated signals contain less and less information and converge to zero. Due to the approximation, the lost information is the local information in the signal while the information kept is the global information. The information lost is called the detail signal [Mal89].

For a complete representation of the signal $f(t)$ at resolution j we need to consider other vector spaces which contain the detail signals up to level j. Consider a signal at level j. Its approximation at the level $j+1$ is equal to the projection of the signal on V_{j+1}. The detail signal at resolution $j+1$ is given by the projection of the signal on the orthogonal complement of V_{j+1} in V_j. Let W_{j+1} be the orthogonal complement:

$$V_j = V_{j+1} \oplus W_{j+1} \tag{3.7}$$

Considering the projection on a further level, then:

$$V_j = (V_{j+2} \oplus W_{j+2}) \oplus W_{j+1} \tag{3.8}$$

It follows that:

$$V_j = V_J \bigoplus_{k=0}^{J-j-1} W_{J-k}, J > j, \quad \forall j \in \mathbb{Z} \tag{3.9}$$

The term multiresolution refers to the simultaneous presence of different resolutions. Eq. 3.9 means that a signal $f(t)$ at a certain resolution at level j can be completely represented or decomposed into an approximation at a lower resolution at the level J and all the details between j and J.

Using the properties of Eqs. 3.2 and 3.3 of $V_j,\ j \in \mathbb{Z}$ it implies:

$$L^2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_j \tag{3.10}$$

Furthermore, the spaces $(W_j)_{j \in \mathbb{Z}}$ inherit the scaling property of Eq. 3.4:

$$f(t) \in W_j \Leftrightarrow f(2^j t) \in W_0, \forall j \in \mathbb{Z} \tag{3.11}$$

To compute the orthogonal projection of a signal $f(t)$ on W_j, an orthonormal basis of W_j is needed. Similar to the vector space V_j, let the set $\{2^{-j/2}\psi_{j,k}(t), k \in \mathbb{Z}\}$ be the orthonormal basis of W_j, where:

$$\psi_{j,k}(t) = \psi(2^{-j}t - k), k \in \mathbb{Z} \tag{3.12}$$

The function $\psi(t) \in L^2(\mathbb{R})$ is called a detail function.

Since the theory for the multiresolution signal decomposition was proposed by Mallat in 1989 [Mal89], the wavelet transform is the most used method to implement the multiresolution transformation. As shown in Section 4.2.2, the wavelet functions are chosen to be the orthonormal bases for the vector spaces W_j and their associated scaling functions to be the orthonormal bases for the approximation vector spaces V_j.

The reconstruction of the original signal without loss of information is only possible if the complete representation is chosen for the transformation of the signal. Corresponding to Eq. 3.9 the reconstruction of the signal in the original resolution can be done at least with the help of the projection of the signal on the vector space $V_J,\ J \in \mathbb{Z}$ at the certain resolution level $j = J$, to form the last approximation, and the detail signals at all levels in the vector spaces W_j, where $1 \leq j \leq J; \{j, J\} \in \mathbb{Z}$.

3.3 Pyramid Tools

3.3.1 Definition

Some early work used the term *pyramids* to describe what *multiresolution analysis* means. In [Ros84a] the author describes the pyramids as data structures that provide successively condensed representations of the information in the input image. The successive levels of the pyramid are reduced-resolution versions of the input image, so that they represent increasingly coarse approximations of the features of the image.

In the following sections, early tools used to create the successive approximations are introduced. Some of these tools do not need the orthogonality condition of the basis functions. Therefore, they were not used for complete representation of the signals.

3.3.2 Average-based Pyramid

The simplest type of an image pyramid is constructed by repeated averaging of the image intensities in non-overlapping 2×2 blocks of pixels. Given an input image of size $2^n \times 2^n$, applying this process yields a reduced image of size $2^{n-1} \times 2^{n-1}$. This image is called the parent image. Applying the process again to the parent image yields a still smaller image of size $2^{n-2} \times 2^{n-2}$ etc. Fig. 3.2 illustrates this process.

As the images are stacked on top of one another, they constitute an exponential tapering "pyramid" of images. In this simple method each node in the pyramid, say k levels above the base, represents the average of a square block of the base of size $2^k \times 2^k$ [Ros84a].

3.3.3 Weighted Average-based Pyramid

The previous method to create lower resolution images is simple and easy to compute. But the sharp cut off characteristic of the unweighted averaging can be undesirable. Overlapping weighted averages would be preferable. However, the next logical step to construct the lower resolution images of the higher analysis levels, is to use a non-overlapping weighted averaging, peaked at the centre of the non-overlapping averaging regions and falling off to zero at their borders.

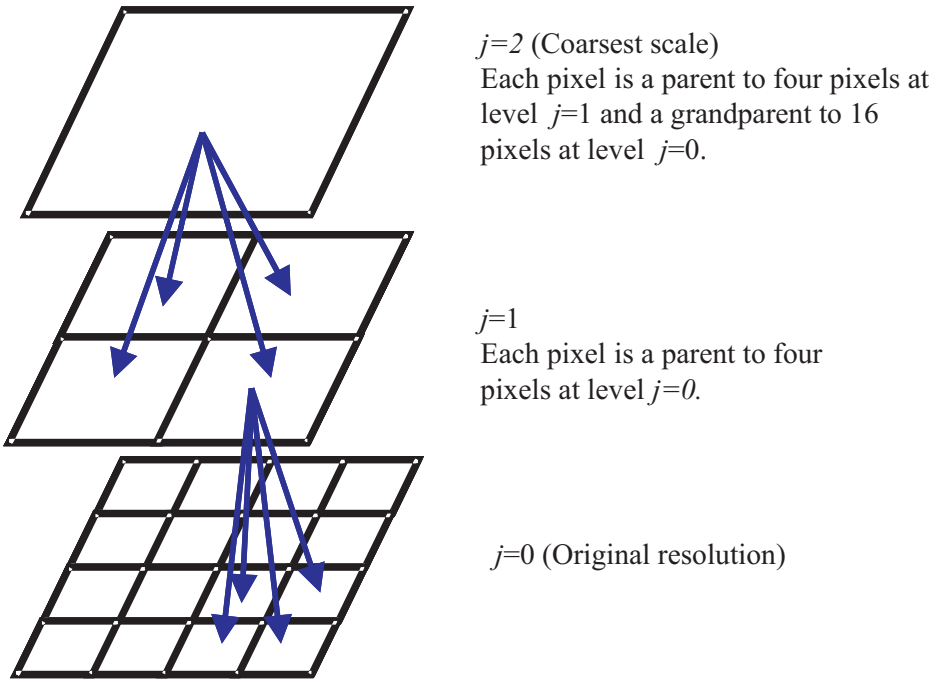


Figure 3.2: Simplest type of an image pyramid.

Salem et al. used in [STMG03] the Gaussian function as a function for the weighted average in a manner similar to a moving window. The original image is divided into parts, each of which has the same size as the filter size. The filter is applied to each part of the image separately. This can be interpreted as a windowed convolution, that also agrees with the concept of a distinct block operation [GW05]. As shown in Fig. 3.3, in the distinct block operation one block of the input image is processed at a time. The operation in this case is Gaussian filtering. Each time the filter is applied on a part of the image, the result is placed as a pixel value in a new image in its corresponding location. For the next images in the pyramid the process is repeated using larger filters. For instance, if the parent image at level $j = 1$ was created with a Gaussian filter of size 3×3 then the grandparent image at level $j = 2$ should be created from the original with a Gaussian filter of the size 5×5. Generally, the distinct block operation may require image padding, since the image is divided into blocks. These blocks will not always fit exactly over the image. In Fig. 3.4 the Gaussian window pyramid is applied to a traffic scene.

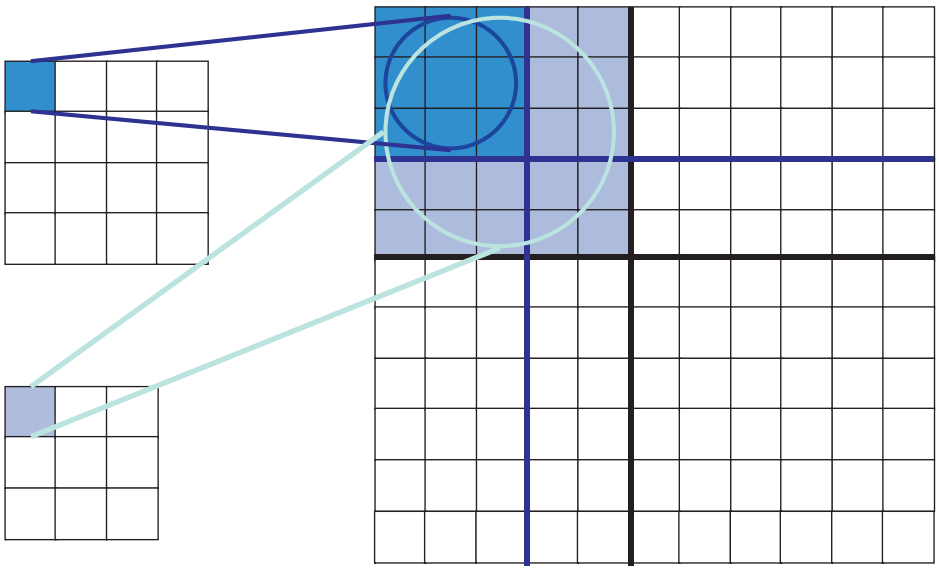


Figure 3.3: Gaussian window for constructing a weighted average pyramid.

In [SAHU04] the authors used the B-spline functions to compute the local weighted geometric moments. A sliding window at dyadic scales is used. The B-splines are well-suited window functions because, in addition to being refinable, they are positive, symmetric, separable, and very nearly isotropic. The algorithm is used in many applications, e.g., as a feature-extraction method for detecting and characterizing elongated structures in images and as a multiscale optical-flow algorithm extending the well-known optical-flow method.

3.3.4 Gaussian Pyramid

The next logical step is to use a weighted and overlapping averaging. Here the functions used to generate approximated version of the signal are non-orthogonal functions. Some redundancy in this method of information representation may be useful. The Gaussian pyramid is a sequence of images, each of which is a low-pass filtered copy of its predecessor [Bur84]. It is called Gaussian pyramid because the low-pass filter used has a Gaussian characteristic.

Let G_0 be the original image. It becomes the bottom or zero level of the Gaussian pyramid. Each pixel of the next pyramid level, image G_1, is obtained as a weighted average of the pixels in image G_0 within an $n \times n$ window.

(a) (b) (c)

Figure 3.4: Image in multiresolution representation. (a) Original resolution. (b) Approximation in one lower resolution level by a 3×3 Gaussian window. (c) Approximation in three lower resolution levels by a 7×7 Gaussian window.

Each pixel of G_2 is then obtained from G_1 by applying the same pattern of weights. The window moves horizontally or vertically so that its centre is the second-next pixel of the current pixel, i.e., the sample distance in each level is double that in the previous level. As a result each image in the sequence is represented by an array which is half as large as its predecessor. The first row of Fig. 3.5 shows an application of the Gaussian pyramid tool to a traffic scene.

3.3.5 Laplacian Pyramid

A set of band-pass filtered images $L_0, L_1, ..., L_{N-1}$ may be defined simply as the differences between the low-pass images at successive levels of the Gaussian pyramid:

$$L_j = G_j - G_{j+1} \tag{3.13}$$

and

$$L_N = G_N \tag{3.14}$$

The image G_{j+1} must be expanded to the size of G_j before the difference is computed. The expansion of an image of size $(M_1 + 1) \times (M_2 + 1)$ is done by interpolating between each two given values to have an image of size $(2M_1 + 1) \times (2M_2 + 1)$. Just as each image in the Gaussian pyramid represents the result of applying a Gaussian filtering to the image of the previous lower level, each image of the set $L_0, L_1, ..., L_{N-1}$ represents the difference of two images of the Gaussian pyramid corresponding to Eqs. 3.13 and 3.14.

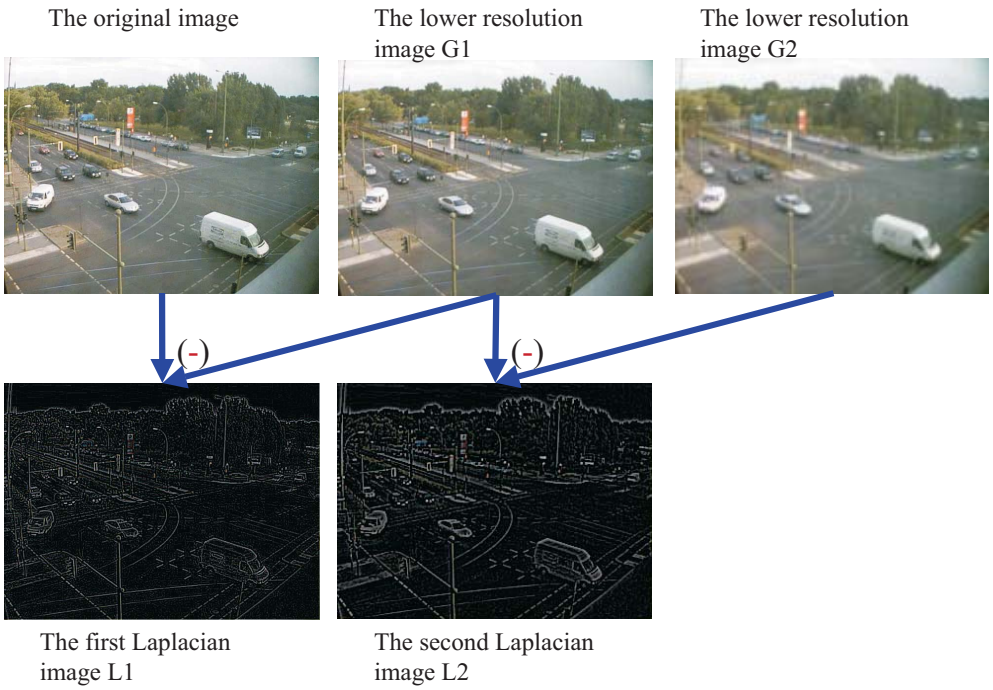


Figure 3.5: Application of the Gaussian (first row) and the Laplacian pyramid tools to a traffic scene.

These differences resemble the Laplacian operator which is used, e.g., in image processing to extract such image features as edges. Therefore, this type of pyramids is called *Laplacian pyramid* [Bur84]. Fig. 3.5 second row shows two levels of a Laplacian pyramid and their relation to the Gaussian pyramid.

3.4 Applications

In many applications it is not required to re-represent the original signal, but to produce a transform of the signal in other spaces or dimensions. The goal is to represent some certain information of the signal in a simpler or more obvious form. In this case only the projection on the approximation vector spaces V_j, $j \in \mathbb{Z}$ is done. Then the approximations are used for further processing.

The multiresolution analysis is proposed for solving many problems in image processing. Most of the selected examples introduced here are based on the Gaussian and the Laplacian pyramids.

3.4.1 Multiresolution Microscopy

In [Pre84] the use of multiresolution image acquisition for human blood-cell analysis is described. To accomplish this a telescope is used instead of a pyramid.

Telescopic processing means: many images are made at various resolutions but always with the same number of pixels. All processing arrays in the telescope are of the same $N \times N$ array size, yielding a varying field of view of the telescope. The actual dimension is determined by the associated image processor. At the highest level of the telescope the images are scanned with the highest desired level of processing or the highest possible sampling.

The blood cells are sampled at a highest sampling rate of 10,000 sample points per mm. Then, the recording is done at 300 sample points per mm, corresponding to about 500,000 sample points. The image is screened at this resolution, and the white cells are located among the red cells. Then the resolution of the telescope is switched to a medium value of 2500 points per mm for further processing. Careful studies have indicated that the images of white blood cells generated at this sampling rate produce satisfactory recognition criteria.

3.4.2 Image Compression

The pyramid representation also permits data compression. If the low resolution images in the pyramid are expanded to the size of the original image as described before, most of the pixel values tend to be near zero. Therefore, they can be represented with a small number of bits. Further data compression can be obtained through quantisation: the number of distinct values taken by samples is reduced by grouping the existing values. This results in some degradation when the image is reconstructed. If the quantisation bins are carefully chosen, the degradation will not be detectable by human observers and will not affect the performance of the analysis algorithms [AAB$^+$84].

In [GY95] the authors used the Gaussian pyramid and the Laplacian pyramid to compress 3D volume data of the computed tomography (CT). A CT data typically consists of 256 slices of 256×256 grey level images with an 8-bit quantised voxel value. The total size of such an image is 16 megabytes.

The images were processed by a $5 \times 5 \times 5$ Gaussian filter to create a pyramid of lower resolution images. The images of the corresponding Laplacian pyramid were uniformly quantised. This reduced the average number of bits per pixel significantly. A compression ratio of 10 : 1 was achieved with a root mean square error of 1.3. Since the 3D Gaussian filter is separable, the low-pass filtering process can be done by using a 1D operator in three passes along the x, y, and z dimensions, respectively. The result is a constant speed up in the reconstruction process by roughly a factor of 8.

3.4.3 Pattern Matching

To locate a particular target pattern that may occur at any scale within an image, the pattern is convolved with each level of the image pyramid. All levels of the pyramid combined contain at most one third more pixels than the original image. Thus, the cost of searching for a pattern at many scales is just one third more than that of searching the original image alone. The complexity of the patterns that may be found in this way is limited by the fact that not all image scales are represented in the pyramid [AAB$^+$84]. As defined here, pyramid levels differ in scale by powers of two. Power-of-two steps are adequate when the patterns to be located are simple. Complex patterns require a closer match between the scale of the query pattern and the scale of the pattern as it appears in the image.

Multiresolution is also used for exemplar-based texture modelling and syntheses [Bon97]. In exemplar-based modelling, sample texture images are used to build statistical texture models for regenerating new instances. That is, to generate a new texture image from an original texture image, such that the new image is sufficiently different from the original one and still appears as if they were generated by the same underlying stochastic process. The author presents an algorithm for texture synthesis, which decomposes the input texture image (the example) into a multiscale pyramid using the standard Laplacian pyramid. A “synthesis pyramid” is generated by sampling the Laplacian pyramid based on parent feature similarity constraints. The new pyramid is used to reconstruct an image which represents the same texture class as the original input image. Texture modelling and synthesis has direct implications for texture recognition, segmentation and classification tasks [SD04].

3.4.4 Image Segmentation

Salem et al. [STMG03] proposed a multiresolution image segmentation for medical magnetic resonance images (MRI) based on the well-known Expectation Maximization (EM) algorithm [TM96], namely: the Gaussian Multiresolution EM algorithm (GMEM). It keeps the advantages of the simplicity of the EM algorithm and overcomes its drawbacks by taking into consideration the spatial correlation between pixels in the classification done afterwards. The neighbouring pixels are spatially correlated because they have a high probability of belonging to the same class. In this work a weighted average pyramid of three successive images is used to utilise the spatial correlation. The Gaussian filter is used without overlapping, where two filters of sizes 3×3 for the parent image and 5×5 for the grandparent image are used.

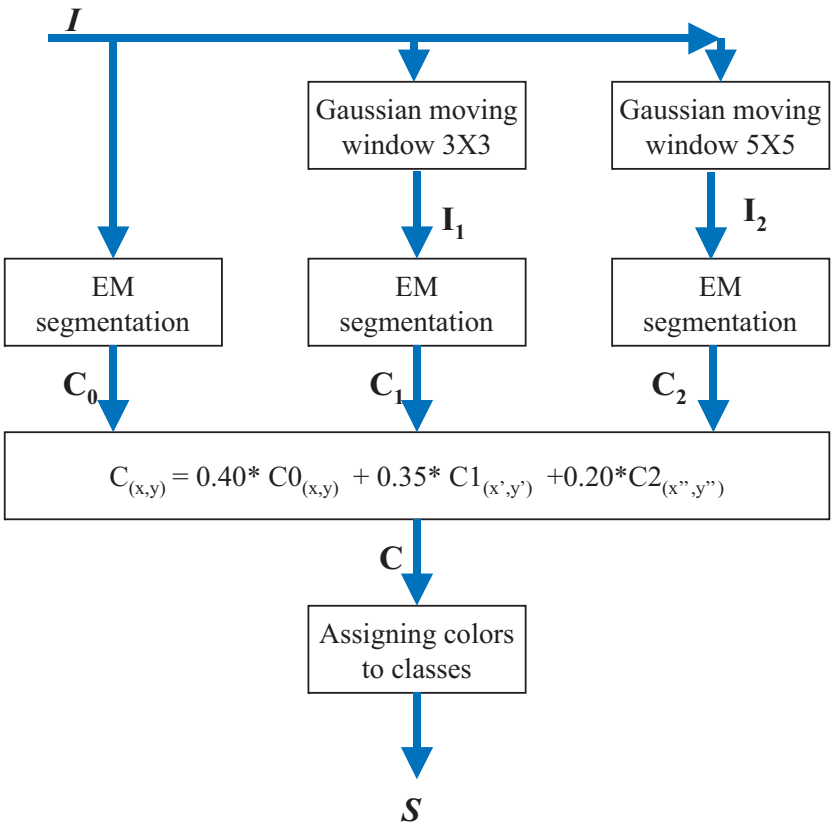


Figure 3.6: Block diagram of the GMEM algorithm. I: input image. S: segmented image.

Once the parent and grandparent images have been created, one moves to the next step by solving the segmentation problem using the different scales of the image. The EM algorithm is used to segment the image at each scale - independent on the others - to produce three segmented images in three successive scales of the original image. The EM algorithm is then followed by a classifier. Three classification matrices C_0, C_1, and C_2 are the output of this step. They represent the results of the segmentation of the original image, its parent, and its grandparent images, respectively. The final step is a weighted combination step done by assigning weights to each classification matrix obtained from the previous step. The GMEM algorithm is summarised in Fig. 3.6. The assigning of the weights reflects the confidence in the segmentation decision of the corresponding level. The weights are chosen such that the following decision roles are ensured:

$$\text{if} \quad C_0(x,y) = C_1(x',y') = C_2(x'',y'') \quad \text{then} \quad C(x,y) = C_0(x,y) \tag{3.15}$$

$$\text{if} \quad C_1(x',y') = C_2(x'',y'') \neq C_0(x,y) \quad \text{then} \quad C(x,y) = C_1(x',y') \tag{3.16}$$

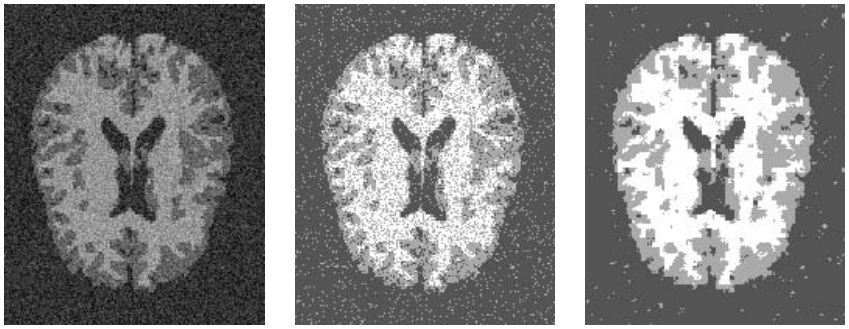
$$\text{if} \quad C_1(x',y') \neq C_2(x'',y'') \quad \text{then} \quad C(x,y) = C_0(x,y) \tag{3.17}$$

where (x', y') is the parent of (x, y) and (x'', y'') is the grandparent of (x, y), and C is the final classification.

The second role in Eq. 3.16 ensures that no pixels in the image will be mistakenly assigned to some class, say class 1, while its parent and grandparent belong to another class, say class 2. The third role in Eq. 3.17 assigns the classification by C_0 to the finial classification if the classification of the parent image C_1 and the grandparent image C_2 are different. This is because the leading weight was assigned to the classification matrix C_0, of the original image. This has to be done if the images contain many edges of high importance. For other types of images where the edges have less importance the greatest weight should be assigned to the classification matrix of the parent image or grandparent image.

The algorithm has been tested using synthetic data and manually segmented magnetic resonance images (MRI). The accuracy of the segmentation increased significantly over that of the conventional EM algorithm. In case of the synthetic data, about 15% increase in the segmentation overall accuracy was obtained for images with much noise. In case of the real MR images with ground truth and added high Gaussian noise, an increase in the segmentation overall accuracy of about 9% is obtained.

The results in Fig. 3.7 show that the new multiresolution algorithm provides superior segmentations over the one-scale image segmentation algorithms. The drawback found in the GMEM algorithm is that the application to pixels laying on the boundaries between classes or on edges generates many misclassified pixels. This is because the parent and grandparent images contain only low frequencies and hence the edges rarely appear in these images. Most of the misclassifications are due to the classification of the parent and grandparent images that were used to reclassify the pixels near the edges.



(a) (b) (c)

Figure 3.7: MRI segmentation. (a) Simulated MRI with added noise. (b) Results by EM. (c) Results by GMEM.

Chapter 4

The Discrete Wavelet Transform

The wavelet transform as an analysis tool plays an important role in the new multiresolution segmentation algorithms described in the following chapters. In this chapter the one- and the multidimensional wavelet transform is introduced. Other points are addressed, such as the Fourier transform and the wavelet packet analysis. They are either fundamentals or required to complete the introduction of the wavelet transform. Moreover, this part of the thesis gives a survey of recent applications of the wavelet transform in image and video segmentation.

4.1 Introduction

To introduce the wavelet transform consider a function

$$f : \mathbb{R}^n \to \mathbb{C}$$

In case $n = 1$, f has only one independent variable t, which is customarily interpreted as the time. In this case $f(t)$ can be called *a temporal signal*, e.g., an audio signal. If $n = 2$, f has two independent variables x and y, customarily they represent a spatial location.

The approximation or the representation of the function f by means of well-known functions ϕ_m is expressed by a linear combination as:

$$f_n = \sum_m c_m \phi_{m,n} \tag{4.1}$$

These well-known functions are usually well understood, easy to compute, and have useful analytical characteristics, such as the polynomials:

$$\phi_m(t) = \sum_{k=0}^{m} a_k t^k, \{\phi : \mathbb{R} \rightarrow \mathbb{R}\} \tag{4.2}$$

The operation of assigning a vector of coefficients c_m to a given function f is called the *analysis* of the function f regarding the family of basis functions ϕ_m:

$$c_m = \sum_n f_n \phi_{m,n} \tag{4.3}$$

The inverse operation that takes a given vector of coefficients c_m as input and returns the function itself as output is called the *synthesis* of f by means of the inverse of ϕ_m.

Wavelets are a system of basis functions that has gained attention in the last two decades because of their characteristics such as orthogonality, compactness, and ease of computing.

Many books start to introduce the wavelet transform by first introducing the Fourier transform. This is done not only to give better understanding of the wavelet transform by showing the similarities and differences from other well-known transforms, but also because the Fourier analysis is the most important tool in the construction of the wavelet theory. Here, only the basic definitions are given so that they can readily be used later on.

The expansion of Eq. 4.1 by means of the complete orthonormal system of the exponential functions

$$\phi_{m,n} = e^{j2\pi \frac{mn}{N}} \tag{4.4}$$

yields the transform:

$$f_n = \frac{1}{N} \sum_{m=0}^{N-1} \hat{f}_m e^{j2\pi \frac{mn}{N}}, \quad n = 0, 1, ..., N-1 \tag{4.5}$$

where N is the number of the sample points, and $1/N$ is the sampling rate for a unit interval.

The analysis of the function f by means of the exponential functions in Eq. 4.4 is given as:

$$\hat{f}_m = \sum_{n=0}^{N-1} f_n e^{-j2\pi \frac{mn}{N}}, \quad m = 0, 1, ..., N-1 \tag{4.6}$$

The last equation is called *Discrete Fourier Transform* (DFT). It generates a discrete spectrum $\hat{f}$ from a discrete input function f. Eq. 4.5 is called *Inverse Discrete Fourier Transform* (IDFT).

The Fourier transform is a mathematical procedure that transforms a time-dependent function f into a new function $\hat{f}$ which depends on the frequency. The Fourier transform treats the input function f as a *one-piece object*. In particular, there is no localisation on the time axis. The value of the function $\hat{f}$ at any frequency point m contains information about f originating from the entire time domain of f. One cannot decide, merely from looking at the $\hat{f}$, where f has, e.g., its maximum or a jump discontinuity.

A function f and its Fourier transform $\hat{f}$ are two representations of the same information. The function f displays the time information and hides the information about the frequencies. The result of the Fourier transform $\hat{f}$ displays information about frequencies and hides the time information. Nevertheless, both functions contain all the information of the signal. Otherwise it would be impossible to reconstruct the function f from its transform $\hat{f}$.

Time localisation can be achieved by first windowing the signal f using a selected window function and then taking its Fourier transform. This technique is called *Windowed Fourier Transform* or *Short-Time Fourier Transform* (STFT). Short-time Fourier transform maps a signal into a two-dimensional function of time and frequency. It is a standard technique for time-frequency localisation. The size of the window function defines the frequency resolution of the process.

Many possible choices have been proposed for the window function in signal analysis, most of which have some compactness and reasonable smoothness. A very popular choice is the Gaussian function. The Gaussian function is supposed to be well concentrated in both time and frequency [Dau92]. The STFT compromises between time and frequency information which is in some applications more useful than the Fourier transform. However, the drawback is that the window is the same for all frequencies. It only depends on the choice of a particular windowing function with a particular size in time.

Many signals require for their analysis a more flexible approach. So one can vary the window size to determine more accurately either time or frequency.

4.2 Wavelet Transform

4.2.1 Foundations

The wavelet transform represents the next logical step: a windowing technique with variable-sized windows. The wavelet transform allows the use of longer windows for extracting more precise low-frequency information from the signal and smaller windows for high-frequency information.

Wavelet transform is a relative recent development in applied mathematics. It is a mathematical tool with a great variety of possible applications. It has already led to exciting applications in signal analysis (sound, images) and numerical analysis, many other applications are being studied. This wide applicability also contributes to the interest the wavelet transform generates. Its name itself was coined approximately less than three decades ago. It is derived to give the meaning "small wave". A wavelet has a waveform of effectively limited duration and has an average value of zero. This local property is the heart of the success of the transform. The foundations given here follow [Dau92, Bla98, GC99, RB00, MMOP07].

Wavelets provide a tool for time and frequency localisation. If one analyses a function $f(t)$ by means of wavelets then there will definitely be some kind of localisation. Transient features (short-time details) of $f(t)$, like jump discontinuities or peaks can easily be localised in the wavelet coefficients of small scales. Long time trends of $f(t)$ are stored in deeper layers of the coefficient hierarchy and are automatically represented in larger scales. As a consequence they are less precisely localised on the time axis.

The wavelet transform has as its goal the analysis and synthesis of functions

$$f : \mathbb{R} \to \mathbb{C}$$

using the function $\psi : \mathbb{R} \to \mathbb{C}$, that satisfies the admissibility condition:

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty \tag{4.7}$$

Then $\psi(t)$ is called a *mother wavelet* or simply a *wavelet*, where $\Psi(\omega)$ is the Fourier transform of $\psi(t)$. C_ψ is a constant that depends on the choice of the wavelet function ψ.

The admissibility condition in Eq. 4.7 implies that:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{4.8}$$

That means that the graph of a wavelet lies partly above and partly below the t-axis and is local.

A set of wavelet functions $\{\psi_{a,b}(t)\}$ can be generated by scaling and translating the mother wavelet $\psi(t)$ by quantities $a,\ b$, respectively as:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{4.9}$$

where $a > 0$ and b are real numbers. The scale factor $1/\sqrt{a}$ ensures that the norm of the wavelet functions remains constant.

The wavelet transform of a function $f(t)$ using the wavelets $\{\psi_{a,b}(t)\}$ can be written as follows:

$$c_{a,b} = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt \tag{4.10}$$

The inverse wavelet transform is the representation of the signal $f(t)$ in terms of the wavelet coefficients $\{c_{a,b}\}$ as:

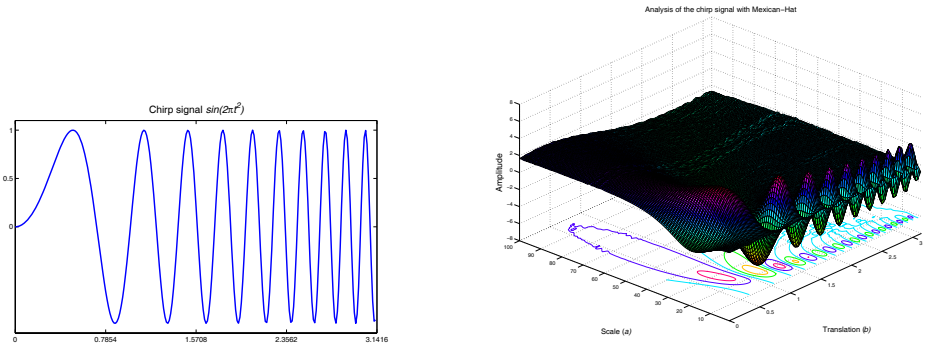
$$f(t) = \frac{1}{C_\psi}\int_a\int_b \frac{1}{a^2}c_{a,b}\psi_{a,b}(t)dbda \tag{4.11}$$

The domain of the coefficients $\{c_{a,b}\}$ in Eq. 4.10 is the (b, a)-plane, i.e., the 2D plane of the set

$$\mathbb{R}^2 = \{(b, a) : b \in \mathbb{R}, a \in \mathbb{R}^*\}.$$

The parameters a and b are called the scale or dilation parameter and the translation or shifting parameter, respectively. The parameter a reflects the scale (width) of a particular wavelet function, while b specifies its translated position along the t-axis.

In case of a *Continuous Wavelet Transform* (CWT) the coefficients are computed for all values of a and b. Fig. 4.1 shows a continuous wavelet transform of a sinusoidal signal shown in Fig. 4.1(a). A coefficient at certain translation b_0 and scaling a_0, as shown in Fig. 4.1(b) represents how closely correlated the wavelet is with this section of the signal.



(a) (b)

Figure 4.1: Continuous wavelet transform. (a) Chirp signal. (b) Wavelet coefficients.

The higher $c_{a,b}$ the more the similarity. More precisely, if the signal energy and the wavelet energy are equal to one, $c_{a,b}$ may be interpreted as a correlation coefficient. Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal.

A flexible local and global analysis is done by changing the values of a and b. In contrast to the Fourier transform changing b gives the ability to analyse a certain part of the signal $f(t)$, while changing a gives the ability in contrast with STFT to change the width of the wavelet and so *implicitly* the frequency bands to be analysed.

Restricting the values of a and b by integers rather than real numbers gives the *Discrete Wavelet Transform* (DWT). However, an interesting type of wavelet transform is to restrict the scaling by factors of two (*binary scaling*), and the translation by an integer multiple of the binary scale factor (*dyadic translation*). Thus, it relates at any scale to the width of the wavelets at that scale, i.e.,

$$a = 2^j, \quad j \in \mathbb{Z}^*$$

and

$$b = k2^j, \quad j \in \mathbb{Z}^*, \quad k \in \mathbb{Z}.$$

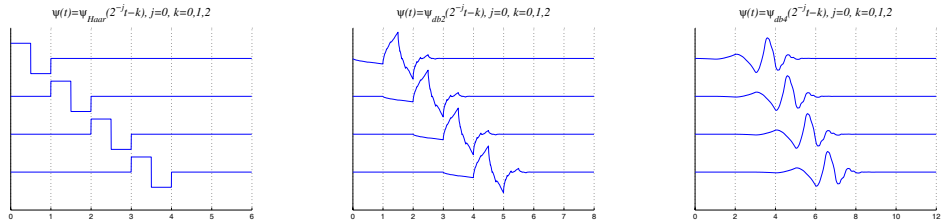
Redefining the set of wavelets of Eq. 4.9 by replacing $a,\ b$ by $j,\ k$ yields:

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k) \tag{4.12}$$

Theoretically, the mother wavelet is progressively scaled by power of two. Each wider wavelet is then translated by increments *approximately* equal to its width, so that the complete set of wavelets at any scale completely covers the analysis interval. Fig. 4.2 shows the dyadic shift operation for three wavelets of the Daubechies family, the wavelets Haar, DB2, and DB4. As the wavelet is scaled up by a power of two, its amplitude is scaled down by powers of $\sqrt{2}$, to maintain the orthonormality. The result of all of this is a set of *orthonormal functions*, which can form an orthonormal basis of $L^2(\mathbb{R})$. Hence, the dyadic wavelet transform for a signal $f(t)$ can be written as:

$$c_{a,b} = \langle f(t), \psi_{j,k}(t) \rangle = \sum_t f(t) 2^{-j/2} \psi(2^{-j}t - k), \quad j, k \in \mathbb{Z} \tag{4.13}$$

The wavelet orthonormal bases provide an important new tool in functional analysis. Before then, it has been believed that no construction could yield simple orthonormal bases of $L^2(\mathbb{R})$ whose elements have good localisation properties in both, the spatial and Fourier domains [Mal89].



(a) (b) (c)

Figure 4.2: Dyadic shifted wavelets. (a) Haar. (b) DB2. (c) DB4.

4.2.2 Wavelet Analysis

Mallat [Mal89] has proposed an iterative algorithm to compute the discrete wavelet transform. Since then this algorithm is the most known method to apply the wavelet transform. It is based on the multiresolution analysis. It applies two bands subband coding procedure in an iterative fashion and builds the wavelet transform from the bottom up, that is, computing small coefficients for small scales first.

The algorithm by Mallat [Mal89] was proposed for signal analysis. He has proposed a mathematical operator which transforms a signal into an approximation at lower resolution, say 2^j. Then he showed that the difference of information between two approximations at resolutions 2^j and 2^{j+1} is extracted by the wavelet function.

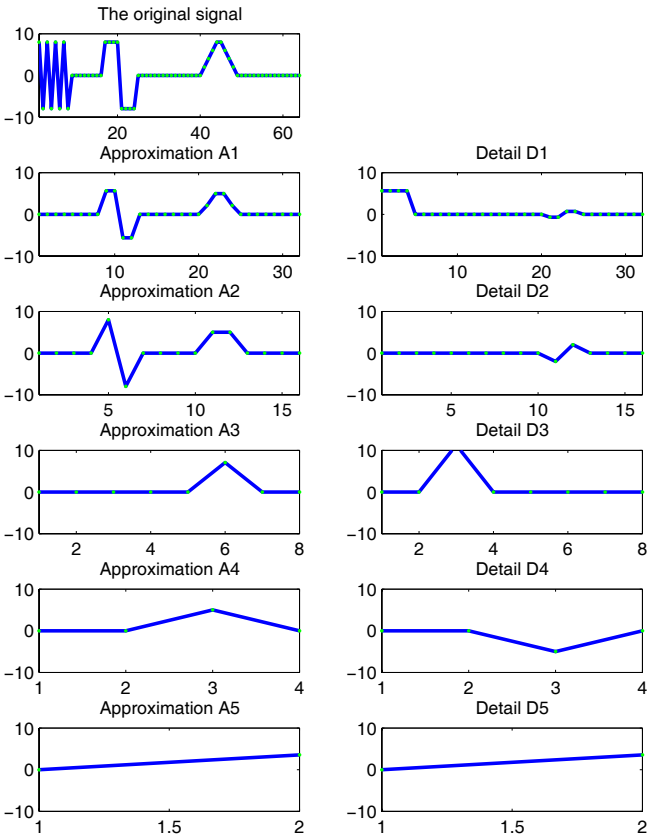


Figure 4.3: Multilevel decomposition using the wavelet transform.

Consider the Fig. 4.3. Every time the function is analysed by the wavelet we go down one level. Certain portions of the function (details) are removed, shown in the right-hand-side plots. Then there are the "approximation" parts, which are further decomposed to give smaller scale representation of the function.

As described in Chapter 3, for a closed approximation vector space $V_j,\ j \in \mathbb{Z}$ a scaling function $\phi(t)$ is defined such that:

$$\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t - k), \quad j,k \in \mathbb{Z} \tag{4.14}$$

An associated function, the *wavelet function*, $\psi(t)$, can be deduced from $\phi(t)$ for the closed *detail* vector spaces $W_j,\ j \in \mathbb{Z}$ such that:

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k), \quad j,k \in \mathbb{Z} \tag{4.15}$$

The families $\{\phi_{j,k}(t)\}$ and $\{\psi_{j,k}(t)\}$ form orthonormal bases for the closed approximation vector spaces $V_j,\ j \in \mathbb{Z}$ and the closed detail vector spaces $W_j,\ j \in \mathbb{Z}$, respectively.

The scaling function ϕ and the wavelet function ψ are related to the low-pass filter H and a high-pass filter G, respectively. The impulse response of H is:

$$h(n) = \langle \phi(t/2), \phi(t-n) \rangle, \quad \forall n \in \mathbb{Z} \tag{4.16}$$

while the impulse response of G is:

$$g(n) = (-1)^{1-n}h(1-n) \tag{4.17}$$

The filter G is the mirror filter of H. G and H are called quadrature mirror filters [Mal89]. As described more fully in Chapter 3 the relation between the approximation and details is:

$$V_j = V_J \bigoplus_{i=j+1}^{J} W_i, \quad \forall j \in \mathbb{Z}$$

Based on this, we can go further on the wavelet representation of the signals, which is shown in the following subsections.

The Approximation

First of all the signal is projected on the approximation vector space $V_j,\ j \in \mathbb{Z}$. As mentioned before the scaled and translated versions of the function $\phi(t)$ at a certain level j_0, i.e., $\{\phi_{j_0}(t)\} = (2^{-j_0/2}\phi(2^{-j_0}t - k)),\ \forall k \in \mathbb{Z}$ form an orthonormal basis of V_j. Then the orthogonal projection on V_j can be computed by decomposing the signal $f(t)$ on the orthonormal basis $\phi_j(t)$, that is $\forall f(t) \in L^2(\mathbb{R})$:

$$A_{j_0} = 2^{-j_0} \sum_{n=-\infty}^{\infty} \langle f(t), \phi_{j_0,n}(t)\rangle \phi_{j_0,n}(t) \tag{4.18}$$

The inner products

$$Ac_{j_0} = (\langle f(t), \phi_{j_0,n}(t)\rangle)_{n\in\mathbb{Z}} \tag{4.19}$$

are called the approximation coefficients of $f(t)$ at the scale j_0.

However, the function $\phi_{j_0,n}$ is a member of the vector space V_{j_0} which is included in the vector space V_{j_0-1}. Hence, it can be projected in the orthonormal basis of V_{j_0-1} by using Eq. 4.18. Let the filter H, with its impulse response h be defined by Eq. 4.16 and consider the expansion of $\phi_{j_0,n}$ on V_{j_0-1}. Eq. 4.19 yields then the approximation coefficients of $f(t)$ that can be re-formulated after Mallat in [Mal89] as follows:

$$Ac_{j_0} = \langle f(t), \phi_{j_0,n}(t)\rangle = \sum_{k=-\infty}^{\infty} \tilde{h}(2n-k)\langle f(t), \phi_{j_0-1,k}(t)\rangle \tag{4.20}$$

where $\tilde{h}$ is the impulse response of the symmetric filter $\tilde{H}$, $\tilde{h}(n) = h(-n)$.

Eq. 4.20 shows that one computes the approximation coefficients Ac_j for any level j by convolving Ac_{j-1} with the filter function $\tilde{H}$ and retains every other sample of the output. Moreover, it shows that the length of the filter is not changed by the change of the scale j, but the resolution of the signal to be filtered is changed by 2^j. This can be formulated as follows:

$$Ac_{j,n} = f(t)\phi_{j,n}(t) = \sum_{k=-\infty}^{\infty} \tilde{h}(2n-k)Ac_{j-1,n} \tag{4.21}$$

All the approximation coefficients Ac_j, $0 < j \leq J$, for some $J \in Z^*$, of a discrete signal $f(t)$ can thus be computed from Ac_0 by repeating this process.

Since H is a low-pass filter, this Ac_j can be interpreted as the result of a low-pass filtering of $f(t)$ followed by a uniform sampling at the rate 2^j [Mal89]. Assuming $j = 1$, then the highest frequency of the signal $f(t)$ is suppressed to one half. Then the sampling rate of the approximation coefficients A_1 must be halved. This is done by retaining every other sample point of the output. This process is called *down sampling*. Due to the convolution with low-pass filters the details of $f(t)$ that have a frequency $\omega > \frac{1}{2} \|\underset{\omega}{\text{argmax}}(\hat{f}(\omega))\|$ are removed.

The Details

To extract the details lost between the approximations of the signal $f(t)$ at the scales $j-1$ and j, the original signal must be projected on W_j, which is the orthogonal complement of V_j in V_{j-1}.

The construction is similar to the one in the approximation. The detail coefficients, Dc_j, are computed at any level j by convolving the signal $f(t)$ with the wavelet function ψ or with the help of the high-pass filter G:

$$Dc_j = \langle f(t), \psi_{j,n}(t) \rangle = \sum_{k=-\infty}^{\infty} \tilde{g}(2n-k) \langle f(t), \phi_{j-1,k}(t) \rangle \tag{4.22}$$

where $\tilde{g}$ is the impulse response of the symmetric filter $\tilde{G}$, $\tilde{g}(n) = g(-n)$.

Eq. 4.22 shows that the details signal Dc_j is computed by convolving Ac_{j-1} with the filter $\tilde{G}$ and retaining every other sample of the output:

$$Dc_{j,n} = f(t)\psi_{j,n}(t) = \sum_{k=-\infty}^{\infty} \tilde{g}(2n-k) Ac_{j-1,n} \tag{4.23}$$

In practice, the signal $f(t)$ is decomposed into two new subband signals using the low-pass filter H and the high-pass filter G yielding the approximation coefficients A_1 and the detail coefficients D_1 at a lower resolution at level $j = 1$. Each of which has as many sample points as the half of the sample points of $f(t)$. The process is repeated on the new approximation coefficients at level j giving the approximation and details at level $j+1$ until a level J is reached. At that level both subbands have only one sample point and the analysis must stop. In fact A_J represents the global approximation of the original signal $f(t) = A_0$ [Kai98].

Thus the signal can be represented *completely* with the sequence:

$$(Ac_J, (Dc_j)_{0<j\leq J}) \tag{4.24}$$

which has the same total number of samples as the original signal $f(t) = A_0$. Fig. 4.4 illustrates this process and shows the customary graphical representation.

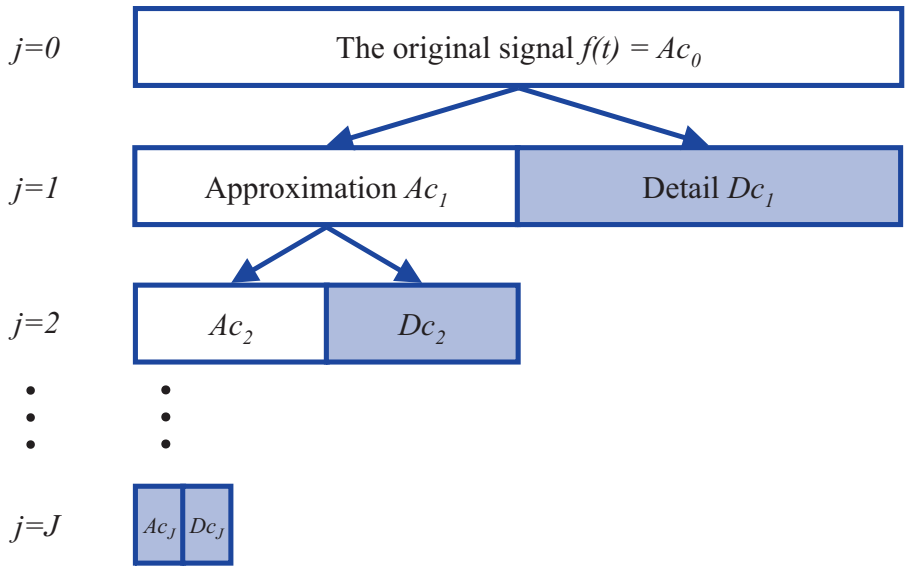


Figure 4.4: Customary representation of 1D dyadic wavelet analysis.

Signal Reconstruction

The reconstruction of the signal can be done by performing the inverse analysis in the inverse direction of the pyramid:

$$Ac_{j,n} = f(t)\phi_{j,n}(t) = \sum_{k=-\infty}^{\infty} h(2n-k)Ac_{j+1,n} + \sum_{k=-\infty}^{\infty} g(2n-k)Dc_{j+1,n} \tag{4.25}$$

The approximation coefficients at a coarser resolution Ac_{j+1} and the signal details at the same resolution Dc_{j+1} are combined together to give the approximation coefficients at level j. The original signal $f(t)$ at resolution $j = 0$ is reconstructed by repeating this procedure for $0 \leq j < J$. Because the reconstructed signal should have a band width as double as each subband of the approximation or details, it should have double the sampling rate and thus double the number of sample points as each one of them. This is done by *up sampling* Ac_{j+1} and Dc_{j+1}, where zeros are inserted between each two sample points.

4.2.3 Wavelet Packet

So far a wavelet decomposition or transform simply re-expresses a function in terms of the wavelet basis $\{\psi_{j,k}(t)\}$. In the case of finite data with information up to a resolution level J, a wavelet transform performs a decomposition of the space V_0 into a direct sum of orthogonal subspaces. Eq. 3.9 yields:

$$V_0 = V_1 \oplus W_1 = V_2 \oplus W_2 \oplus W_1 = ... = V_J \bigoplus_{k=0}^{J-j-1} W_{J-k}, J > j, \quad \forall j \in \mathbb{Z}$$

This "splitting trick" or splitting algorithm can be used to decompose W_j spaces as well. It has been shown [Dau92] that one can define a new set of orthonormal functions from $\{\psi(t)\}$ and $\{\phi(t)\}$ to form new bases, so that this splitting algorithm takes place not only on V_j but also on W_j, $j \in \mathbb{Z}$. This new function bases are called a *library of wavelet packet bases.*

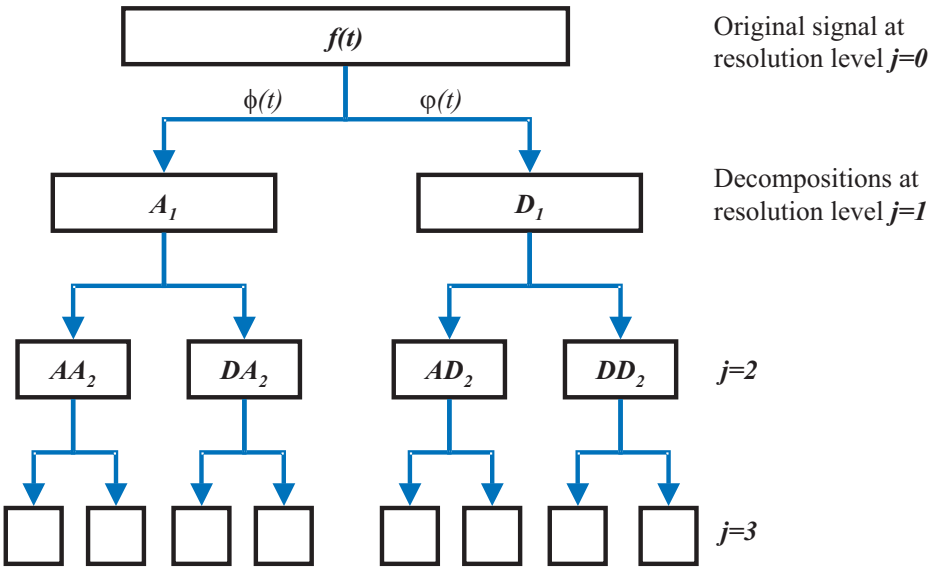


Figure 4.5: Three level wavelet packet decomposition tree.

In wavelet decomposition one leaves the high-frequency part alone and keeps splitting the low-frequency part. In wavelet packet decomposition, we can choose to split the high-frequency part also into a low-frequency part and a high-frequency part. So in general, all subbands are subject for further decomposition. The wavelet packet decomposition divides the frequency space into various parts and allows better frequency localisation of signals. Fig. 4.5 shows a wavelet packet decomposition tree up to the third level.

As in the wavelet transform, one can keep doing the decomposition until a scalar value. On the other hand, one can also choose not to decompose a particular subspace while decomposing others. So there are many choices for decomposing a signal. One can keep all the coefficients at all decomposition levels and generate a table of coefficients of wavelet packet decomposition [CMQW93].

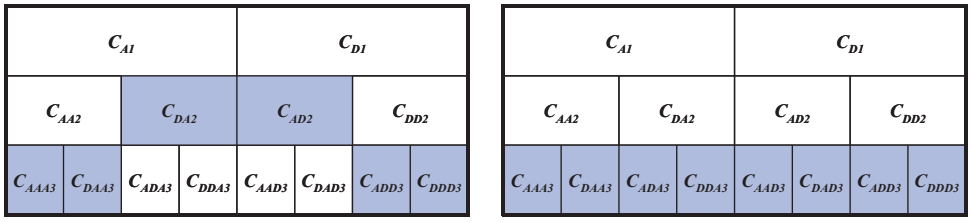


Figure 4.6: Wavelet packet's table of coefficients. The shaded coefficients are two examples for a complete representation of the signal.

This table of coefficients is highly redundant and one needs to choose among all the representations the one that represents the signal most efficiently. For a j-level decomposition there are more than $2^{2^{j-1}}$ different ways to decompose a signal. Fig. 4.6(a) shows selected subbands to represent the original signal. The commonly used criterion for choosing the most efficient or best basis for a given signal is the minimum entropy criterion [CMQW93, MMOP07]. In general, the smaller the entropy the fewer significant coefficients are needed to represent the signal.

Usually the chosen coefficients are shown as shaded boxes in the coefficients table. Sometimes one may prefer to choose basis functions that are all in the same level. Then the question here is to choose the best level and the chosen coefficients may be as shown in Fig. 4.6(b).

4.2.4 Family of Daubechies Wavelets

Daubechies constructed the first wavelet family of scale functions that are orthogonal and have finite vanishing moments, i.e., compact support [Dau92]. This property insures that the number of non-zero coefficients in the associated filter is finite.

The compact support property is very useful for local analysis. This means that the analysis wavelet ψ has zero values outside a certain domain U, which is contained within a circle of radius ρ: $\psi(u) = 0, \forall u \notin U$. Then the wavelet ψ is localised. The signal $f(t)$ and the wavelet ψ are then compared within the circle, using only the t values within the circle. The signal values, which are located outside of the domain U, do not influence the value of the coefficient:

$$\int_{\mathbb{R}} f(t)\psi(t)dt = \int_{U} f(t)\psi(t)dt$$

At a certain position b the corresponding coefficient $c_{a,b}$ analyses $f(t)$ around b. Thus, this type of analysis is local. Not every wavelet has a compact support. This is the case, for instance, for the Meyer wavelet [MMOP07].

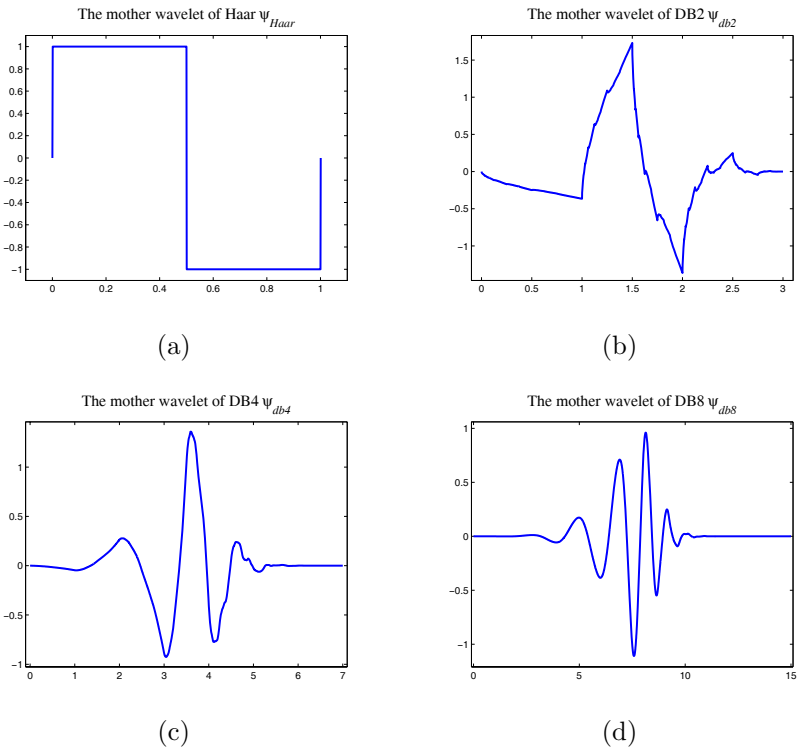


Figure 4.7: Mother wavelet functions $\psi(t)$. (a) Haar. (b) DB2. (c) DB4. (d) DB8.

The Haar wavelet ψ_{haar}, as shown in Fig. 4.7(a), is the basis of the simplest wavelet transform. Historically, it is the first mention of what is called now "wavelet" in the thesis by Alfred Haar in 1909. It is discontinuous and resembles a step function. The Haar wavelet transform has the advantages of being conceptually simple, fast and memory efficient, since it can be calculated in place without a temporary array. On the other hand, the Haar transform has limitations because it is discontinuous, which can be a problem for some applications, e.g., audio signal compression. It is the only symmetric wavelet in the Daubechies family and the only one of them that has an explicit expression. The associated filter is of length two. This means that the resulting approximation and details are all half the number of columns and rows of the input signal. The scale function ϕ_{haar} is the simple average function and the wavelet ψ_{haar} is the difference.

For higher order Daubechies wavelets ψ_{dbN}, N denotes the order of the wavelet and the number of the vanishing moments. The regularity increases with the order, as shown in Fig. 4.7. The support length of ψ_{dbN} and ϕ_{dbN} is $2N-1$. The length of the associated filter is twice as the number of the vanishing moments, i.e., $2N$. The approximation and detail coefficients are of length $\mathsf{floor}(\frac{n-1}{2})+N$, if n is the length of $(f(t))$ [Mal89, Dau92].

The wavelets with fewer vanishing moments give less smoothing and remove less details, but the wavelets with large vanishing moments produce distortions [KT05].

The scaling functions associated with the wavelets of Fig. 4.7 are displayed in Fig. 4.8. Daubechies has designated compactly supported scaling functions first to ensure that the associated wavelets are also compactly supported. She has derived a relation between ψ and ϕ using the multiresolution properties of Eqs. 3.4 and 3.6 so that:

$$\psi(t) = \sum_n (-1)^n \alpha_{-n-1} \phi(2t-n) \tag{4.26}$$

where $\alpha_n = \sqrt{2}\langle \phi, \phi_{1,n} \rangle$, so that $\phi(t) = \sum_n \alpha_n \phi(2t-n)$.

The analysis is done with some overlapping depending on N, since the length of the associated filter is $2N$, and because the translation k of the wavelet during the dyadic analysis is done in terms of the scale level $j \in \mathbb{Z}$ and not of the number of vanishing moments N of the wavelet, i.e., $k = 2^j b, b \in \mathbb{Z}$.

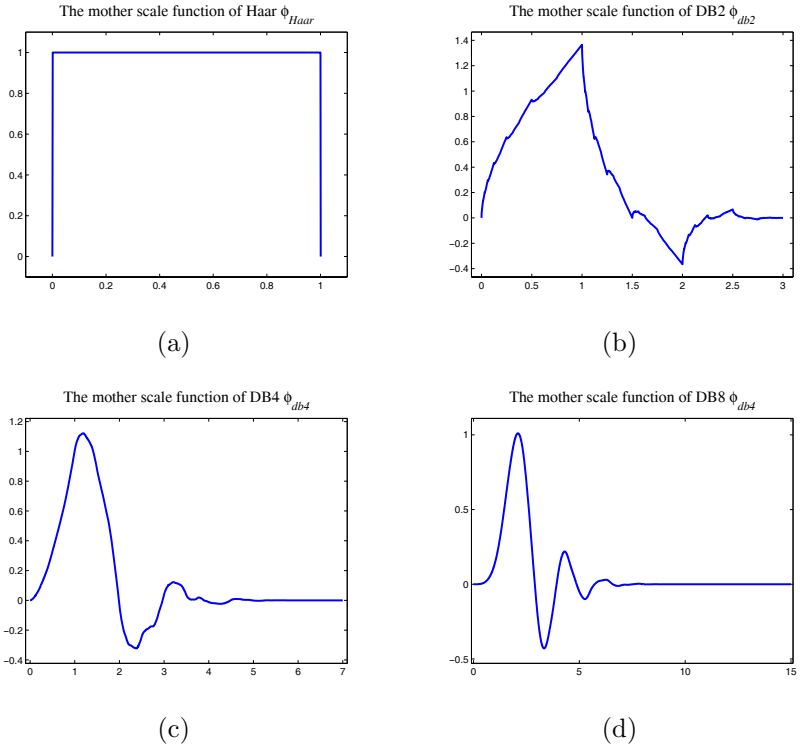


Figure 4.8: Scaling functions $\phi(t)$ associated with the wavelets (a) Haar. (b) DB2. (c) DB4. (d) DB8.

For the Haar wavelet ψ_{haar} there is no overlapping at all because the translation step is equal to the width of the wavelet. However, for the wavelet ψ_{db2} the overlapping is equal to 2 sample points. Generally, for a wavelet of order N at scale j the length of the associated filter is constant, as shown in Eqs. 4.21 and 4.23, and the overlapping is equal to $2^j(N-1)$ sample points.

Concerning the last discussion the detection of an event by higher order wavelets takes longer than that by lower order. Fig. 4.9 shows a 1D signal that has three main events. The first event appears in the beginning of the signal in a form of a zigzag line. The second one represents three sharp edges. The last event represents a wide edge with a slow transition. The detail coefficients of two levels of analysis show that ψ_{haar} has the ability to localise the event even in higher levels.

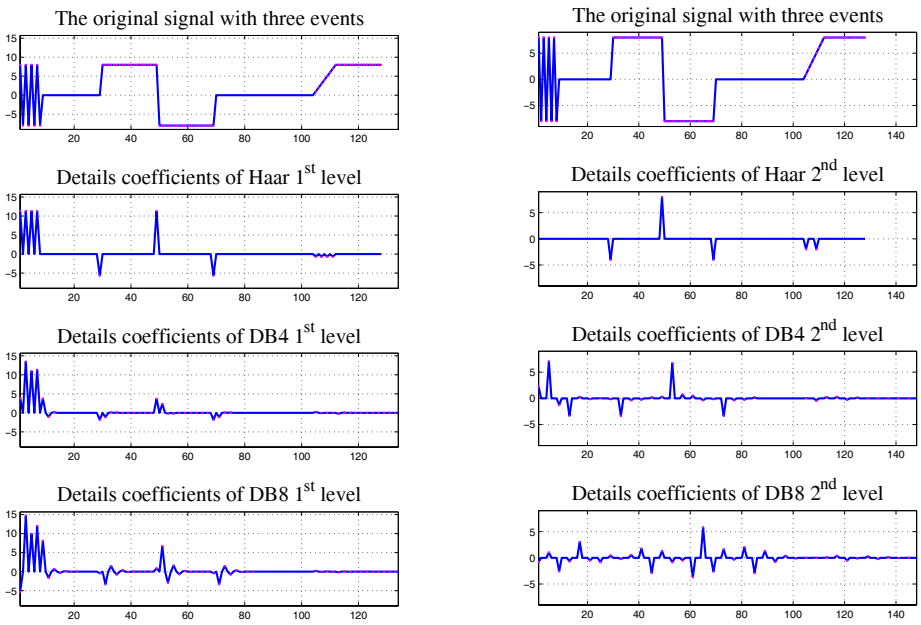


Figure 4.9: Detail coefficients for a two level analysis using ψ_{haar}, ψ_{db4}, and ψ_{db8}. The ψ_{haar} detects all the rapid changes in the first level, while the detection of the wide edge lasts longer. The detection of events in ψ_{db4} and ψ_{db8} comes shifted in position and distributed on a wider range.

4.3 2D Discrete Wavelet Transform

The fast wavelet transform as introduced by Mallat [Mal89] uses separable orthogonal basis functions. Therefore, the multidimensional transform can be decomposed into a tensor product of orthogonal subspaces. This way the n-dimensional transform is computed by n *one*-dimensional convolutions, one in each dimension.

The 2D wavelet transform is widely used for analysis and processing of images and videos. This transform is performed by two separate 1D transforms along the rows and the columns of the image data constructing one 2D scaling function and three different 2D wavelet functions.

The 2D dyadic scaling function ϕ and wavelet functions ψ^h, ψ^v and ψ^d can be expressed as follows:

$$\phi_{j,\{k,l\}}(x,y) = \phi_{j,k}(x)\phi_{j,l}(y) \qquad = 2^j\phi(x-2^{-j}k)\phi(y-2^{-j}l) \qquad (4.27)$$

$$\psi^h_{j,\{k,l\}}(x,y) = \phi_{j,k}(x)\psi_{j,l}(y) \qquad = 2^j\phi(x-2^{-j}k)\psi(y-2^{-j}l) \qquad (4.28)$$

$$\psi^v_{j,\{k,l\}}(x,y) = \psi_{j,k}(x)\phi_{j,l}(y) \qquad = 2^j\psi(x-2^{-j}k)\phi(y-2^{-j}l) \qquad (4.29)$$

$$\psi^d_{j,\{k,l\}}(x,y) = \psi_{j,k}(x)\psi_{j,l}(y) \qquad = 2^j\psi(x-2^{-j}k)\psi(y-2^{-j}l) \qquad (4.30)$$

The results of the analysis at each decomposition level are a low-pass image or a coarser approximation A and three detail images, horizontal details H, vertical details V, and diagonal details D, which contain the details lost while going from the original image to its approximation A. Fig. 4.10 shows a customary representation of the outputs of the 2D analysis.

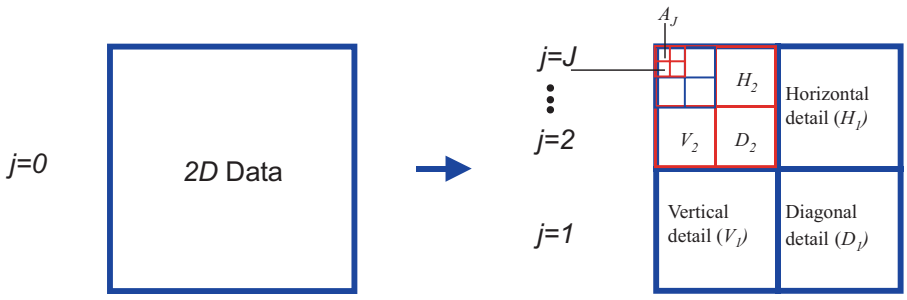


Figure 4.10: Customary representation of 2D dyadic wavelet analysis.

The approximation A represents the image at a coarser resolution. It results from averaging the image in both dimensions x and y. The horizontal detail H is obtained by averaging in the x-dimension and differencing in the y-dimension. The vertical detail V is obtained by averaging in the y-dimension and differencing in the x-dimension. The diagonal detail D is obtained by differencing in both dimensions and then averaging. As shown in Fig. 4.11 horizontal edges tend to show up in H and vertical edges in V, while D contains all other details [Kai98].

The three detail images H, V, and D, can be combined to create a new image that contains only the edges of the original image. It is possible that some edges do not appear in the first level of analysis. In this case further analysis levels must be produced.

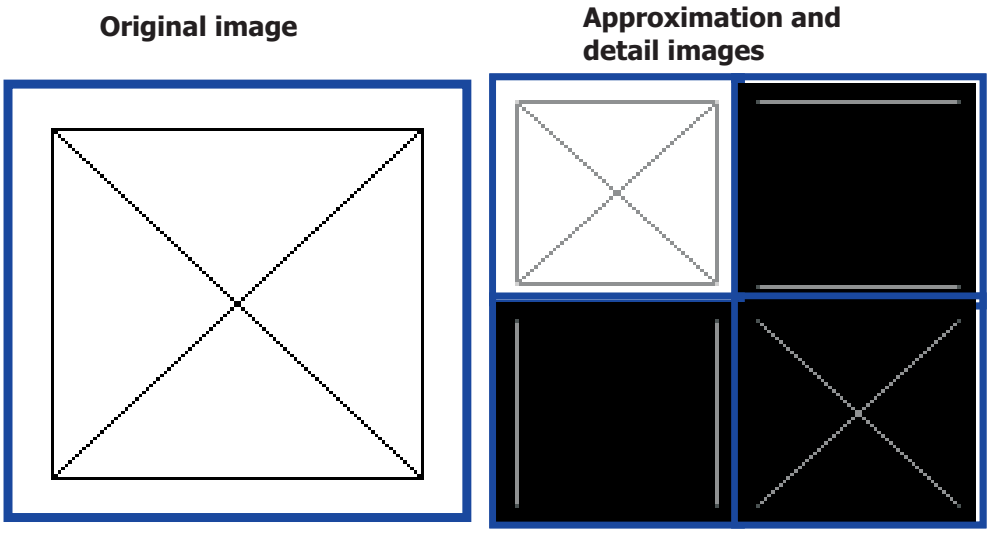


Figure 4.11: Approximation and details of an image.

4.4 3D Discrete Wavelet Transform

4.4.1 Decomposition Schemes

The 3D scaling function and the 3D wavelet functions can each be expressed as a product of three *one*-dimensional functions. The analysis is carried out along the x-dimension, the y-dimension, and the z-dimension of the volumetric data. Eight coefficients result from the one level analysis. One coefficient represents a volume approximation of the input data. The information which is missed in the approximation is distributed in the other 7 volume detail coefficients. Fig. 4.12 shows a one level 3D transform done as three stand-alone 1D transforms.

The 3D dyadic scaling function ϕ and the wavelet functions $\psi^i, i = 1, 2, ..7$ can be expressed as follows:

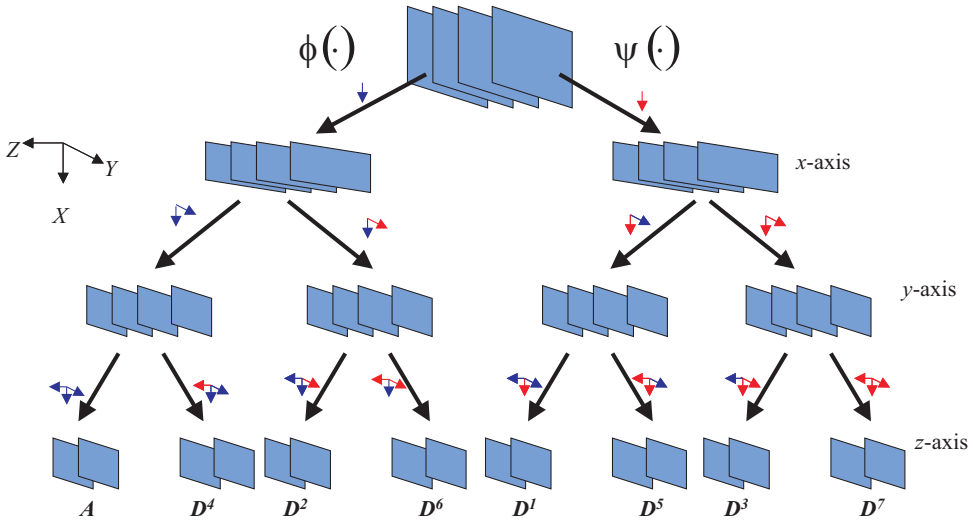


Figure 4.12: 3D Wavelet transform as three 1D wavelet transforms.

$$\phi_{j,\{k,l,m\}}(x,y,z) = 2^{\frac{3j}{2}}\phi(x-2^{-j}k)\phi(y-2^{-j}l)\phi(z-2^{-j}m) \tag{4.31}$$
$$\psi^1_{j,\{k,l,m\}}(x,y,z) = 2^{\frac{3j}{2}}\psi(x-2^{-j}k)\phi(y-2^{-j}l)\phi(z-2^{-j}m) \tag{4.32}$$
$$\psi^2_{j,\{k,l,m\}}(x,y,z) = 2^{\frac{3j}{2}}\phi(x-2^{-j}k)\psi(y-2^{-j}l)\phi(z-2^{-j}m) \tag{4.33}$$
$$\psi^3_{j,\{k,l,m\}}(x,y,z) = 2^{\frac{3j}{2}}\psi(x-2^{-j}k)\psi(y-2^{-j}l)\phi(z-2^{-j}m) \tag{4.34}$$
$$\psi^4_{j,\{k,l,m\}}(x,y,z) = 2^{\frac{3j}{2}}\phi(x-2^{-j}k)\phi(y-2^{-j}l)\psi(z-2^{-j}m) \tag{4.35}$$
$$\psi^5_{j,\{k,l,m\}}(x,y,z) = 2^{\frac{3j}{2}}\psi(x-2^{-j}k)\phi(y-2^{-j}l)\psi(z-2^{-j}m) \tag{4.36}$$
$$\psi^6_{j,\{k,l,m\}}(x,y,z) = 2^{\frac{3j}{2}}\phi(x-2^{-j}k)\psi(y-2^{-j}l)\psi(z-2^{-j}m) \tag{4.37}$$
$$\psi^7_{j,\{k,l,m\}}(x,y,z) = 2^{\frac{3j}{2}}\psi(x-2^{-j}k)\psi(y-2^{-j}l)\psi(z-2^{-j}m) \tag{4.38}$$

Using Eqs. 4.31 to 4.38 the 3D analysis gives the following 8 subbands:

$$\begin{aligned} A_{j,\{k,l,m\}} &= \langle f(x,y,z),\ \phi_{j,\{k,l,m\}}(x,y,z)\rangle, \\ D^i_{j,\{k,l,m\}} &= \langle f(x,y,z),\ \psi^i_{j,\{k,l,m\}}(x,y,z)\rangle \end{aligned} \qquad \begin{aligned} j \in \mathbb{Z}, \forall\{k,l,m\} \in \mathbb{Z}^3, \\ i = 1,..7 \end{aligned} \tag{4.39}$$

In Eq. 4.39 A_j is the low-pass subband at resolution level j and D_j^i is the high-pass subband i at resolution level j. For the fast 3D wavelet analysis only the low-pass subbands $A_{j,\{k,l,m\}}, j \in \mathbb{Z}, \{k,l,m\} \in \mathbb{Z}^3$ are used for further decomposition at lower resolution levels. Therefore, it requires only $\mathcal{O}(n)$ computations [Dim02]. The subband A is generated applying the scaling function ϕ to all the three dimensions of the data. So it represents the approximation in all axes. It can be used to replace the original data if no relevant changes occur. The subbands D^1, D^3, D^5, and D^7 are the result of applying the wavelet function ψ on the x-axis. So all of them contain information about the possible changes along the x-axis. Therefore, information about vertical edges can be easily obtained from such subbands. The subbands D^2, D^3, D^6, and D^7 and the subbands D^4, D^5, D^6, and D^7 can be explained in the same way as gradient information along the y-axis and z-axis, respectively.

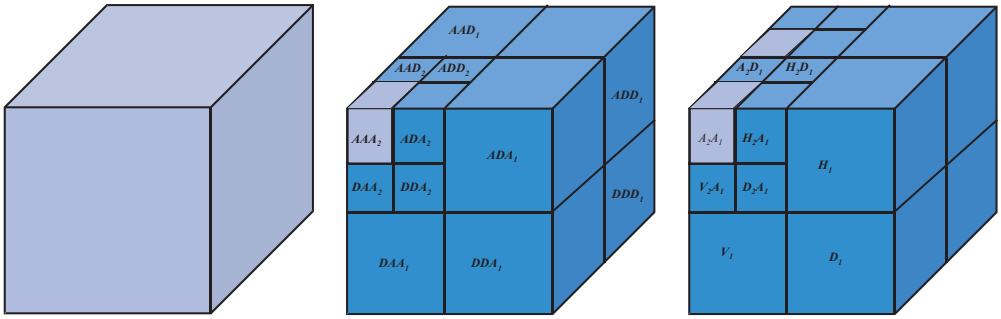
The notations of A and $D^i, i = 1,..7$ are used to represent the subbands resulting from the transform, while the notations AAA, DAA, ADA, DDA, AAD, DAD, ADD, and DDD are used to represent instant images in the subbands A and $D^i, i = 1,..7$, respectively.

For the conventional 3D transform the input is a cubic shape data element with three dimensions, width, height, and depth. In this case the smallest number of sample points is eight, two sample points along each dimension. For multiple level analysis the input data must be at least of size 2^{3j} where $j \in \mathbb{Z}$ is the desired analysis level. Fig. 4.13(a) shows a cubic data and Fig. 4.13(b) the corresponding transform using two level 3D wavelet analysis. The complete representation of a 1D signal in Eq. 4.24 can be extended to 3D:

$$\left(A_J,\ (D_j^i)\right), \quad (i = 1,..7;\ 0 < j \leq J) \tag{4.40}$$

Each coefficient has a cubic shape, and the whole sequence has the same number of sample points as the original data.

For many applications this decomposition scheme is preferable because it offers simple and straightforward data synthesis. Data synthesis is done in similar manner as the composition scheme introduced by Eq. 4.25.



(a) (b) (c)

Figure 4.13: Transforming (a) 3D data using (b) Two levels of 3D wavelet. (c) Two levels 2D wavelet + one level 1D wavelet.

4.4.2 2D+1D Discrete Wavelet Transform

Alternative to the conventional 3D decomposition one can decompose the volumetric data first by the 2D wavelet transform (spatially) for any arbitrary number of levels m, followed by the 1D transform (temporally) for any other arbitrary number of levels n. This decomposition scheme is known as a *3D wavelet packet transformation* [FR07]. For example, the wavelet packet transform as illustrated in Fig. 4.13(c) uses a two level decomposition ($m = 2$) spatially and one level decomposition ($n = 1$) temporally to analyse a 3D data set. The 3D data set, for instance an image sequence, is first analysed using the 2D wavelet transform for two levels. All coefficients of the results of the last analysis level (A_2, H_2, V_2 and D_2) are then used to be analysed by the 1D wavelet analysis in the temporal domain for one level. The results of the last level are eight coefficients equivalent to the eight coefficients of the conventional 3D wavelet analysis. Fig. 4.14 illustrates this process.

To have a complete representation of the original data, all the spatial details for the first level must be saved in addition to the details and the approximation of the temporal analysis.

4.5 Image Segmentation Applications

Since the appearance of the wavelet transform two decades ago it has been widely used in many applications in image processing. Visual data exhibit two important characteristics related to each other. They consist of signals with many different scales or frequencies, and these signals have spatially inhomogeneous magnitudes.

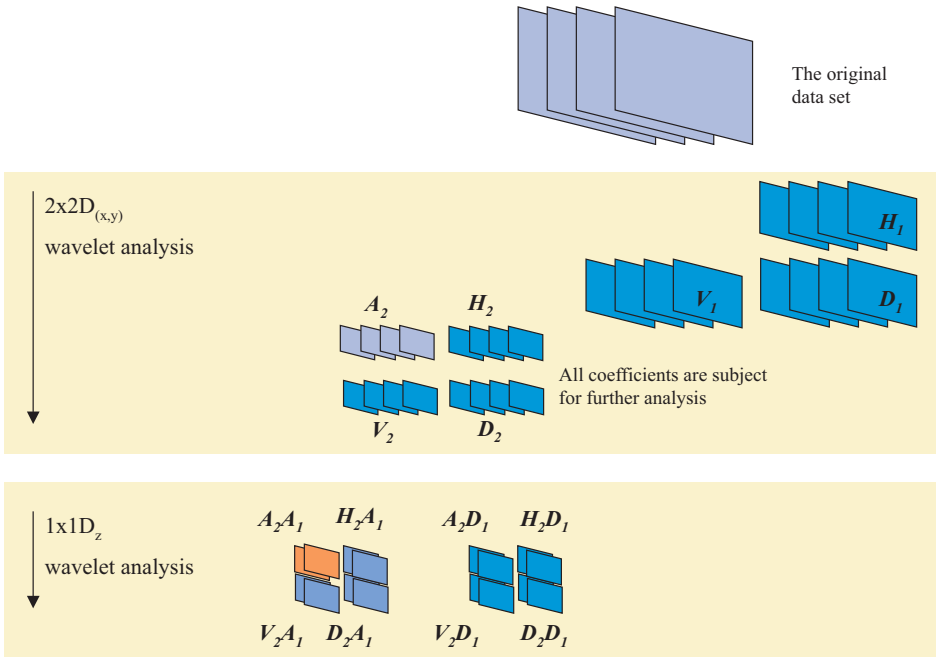


Figure 4.14: 2D + 1D wavelet packet for analysis of image sequences.

The wavelet transform has successfully exploited both characteristics to achieve a transformation that is local in both the frequency and the space domains. As a result the structures of the original signal become better exposed to analysis. The wavelet analysis has been frequently applied to image compression [Kai98, IP99], feature extraction for the purpose of image indexing, matching and content based image retrieval as in Salem et al. [TSNEA05], and for medical image segmentation [SRKN98, SMTG01]. In video coding and compression as well as in video transition [SVMJ95, MQG+01, EDS+05]. However, we will focus in this part on the recent work in image segmentation based on the wavelet transform.

The Wavelet Multiresolution Expectation Maximization (WMEM) algorithm, proposed by Salem et al. in [SMTG01], is a segmentation algorithm that uses the EM algorithm and the multiresolution analysis to address two aspects. First, the image subject to segmentation can be modelled as a statistical mixture model with missing parameters. That is, the pixel intensities are the incomplete data (known a priori) and the missing data are the class of the pixels. Second, other features than the pixel intensity must be used by the segmentation algorithm.

The WMEM algorithm uses two levels Haar wavelet analysis to create two lower resolution images. The approximation of the first level is used as a parent image A_1, while the horizontal, H_1, vertical V_1, and diagonal D_1 details are used to create an edge image. This edge image is called a *mask* M_1. The same process is done for the second level of analysis to have A_2 as a grandparent image and its mask image M_2.

To create a mask, the details H, V, and D are converted into binary images by choosing a suitable threshold for each of them. Then these three binary images are combined in one image by performing the logical OR operation. A pixel's value in the mask image can be either 1 or 0 depending on whether the corresponding pixel in the approximation image is generated from pixels laying on an edge or not.

The conventional EM algorithm is then applied on the input image I, the parent image A_1 and the grandparent image A_2. The EM algorithm is followed by a classifier, so the outputs of this step are the classification matrices C_0, C_1, and C_2.

The WMEM is based on the assumption used in building the GMEM algorithm presented in Section 3.4.4. This assumption is that the classification of a pixel at some resolution j is dependent upon the classification of its parent at resolution $j+1$ and the classification of its grandparent at resolution $j+2$. Each pixel is then reclassified using the classification matrices C_0, C_1, and C_2 after assigning a weight to each of them. The weights have been assigned under the conditions given in Eqs. 3.15, 3.16, and 3.17.

However, since the pixel in the parent image is computed from four neighbour pixels this pixel could be generated from different classes. This will in most cases affect the classification of this pixel negatively. The same is true for the pixels of the grandparent image. For this reason, when we use the classifications of the parent or grandparent pixels we have to be sure that those pixels are not a mixture of different classes. Since the mask images M_1 and M_2 do not contain anything except the edges of the associated low resolution images they can be used to prevent those pixels from being used in the reclassification step. The reclassification step is modified so that it does not take place unless the pixels of the parent and grandparent images are not appearing in their associated masks.

The WMEM algorithm is summarised in the flowchart shown in Fig. 4.15. For the "EM segmentation" step in the flowchart please refer to Section 2.2.2.

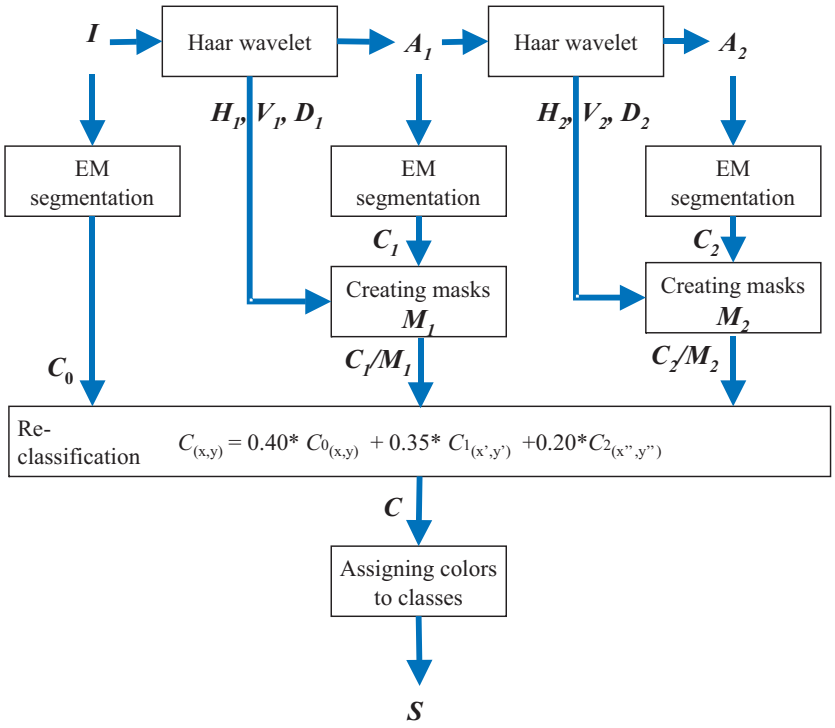


Figure 4.15: Block diagram of the WMEM algorithm. I: input image. S: segmented image.

The authors in [KK07] used the shift-invariant discrete wavelet transform (SIDWT) proposed by Beylkin [Bey92] to extract textural features which describe the relative homogeneity of localised areas of retinal images. The objective of the work was to classify automatically various pathologies from normal retinal images. There was a test data set supplied with ground truth of 38 normal and 48 abnormal images. The SIDWT was used to decompose the images up to the fifth level. From the wavelet coefficients a combination of homogeneity features from the fourth decomposition level with entropy from the first, second and fourth decomposition levels are selected for the classification process. Afterwards the classification is done using the linear discriminate analysis (LDA). The classification results are displayed in a confusion matrix, where specificity of 79% and sensitivity of 85.4% were achieved.

In [Gua06] the author developed an algorithm for lip extraction from colour images based on wavelet multiscale edge detection. First, the image was subject to enhancement using a 3×3 Gaussian filter.

The RGB images were then transformed using a 3×3 discrete Hartley matrix to uniform the colour system. Only the resulting coefficient $C3$ is used for the following multiscale edge detection using the wavelet transform. Based on the resulting edge image the Brunelli method [BP93] is used to localise the lip region. Fig. 4.16 illustrates the process.

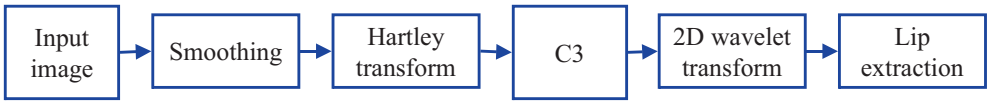


Figure 4.16: Block diagram for lip extraction based on 2D wavelet corresponding to the method of [Gua06].

In [BW06] the authors used the 2D discrete wavelet transform for edge detection. An input grey image is transformed up to the 6^{th} analysis levels. A binary edge image corresponding to each level is created, where the detail subbands images are combined and then thresholded. The experimental results include testing different mother wavelets such as the Daubechies family and the bi-orthogonal spline wavelets and several combination methods of the different levels. The best results are found using the auto-correlation method to combine the edge images of the first three levels.

In [STS07] the authors developed an algorithm for the detection of connected and disconnected boundaries in an image, so that it incorporates noise elimination. The algorithm favours boundaries that exist at multiple resolutions and suppresses boundaries that exist at fine resolution. The directional boundaries are taken at various resolutions up to the fourth analysis level, and are then combined to form a boundary image. Each of the detail coefficients is thresholded and scaled to the original size of the image. It is then multiplied by the four other coefficients in the concordant detail subband. The three resulting images are added together to obtain the augmented boundaries of the image. Fig. 4.17 demonstrates the process. Different wavelets were tested for geometric and natural scene grey level images. Using the Peak Signal to Noise Ratio (PSNR) as an error measure, the best results were obtained by the Haar wavelet.

Chen et al. [CLL02] used the wavelet transform for accurate optical flow estimation. The wavelet is used to approximate both the flow vectors as well as the image related operators. Each flow vector and each image function were represented by linear combinations of wavelet basis functions.

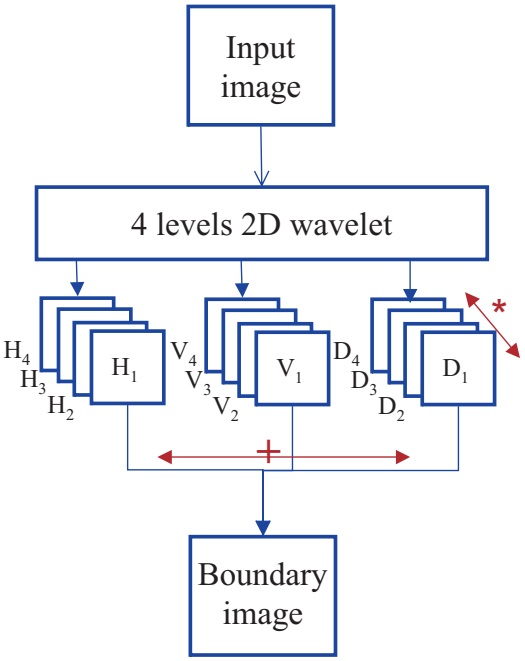


Figure 4.17: Block diagram for boundary detection using 2D wavelet transform corresponding to the method of [STS07].

For an image of size $M \times N$, the flow vectors ($u(x, y)$ and $v(x, y)$) introduced in Eq. 2.24 can be written as follows:

$$u(x, y) = \sum_{m \in M} \sum_{n \in N} u_{m,n} \phi(x - m, y - n) \tag{4.41}$$

and

$$v(x, y) = \sum_{m \in M} \sum_{n \in N} v_{m,n} \phi(x - m, y - n) \tag{4.42}$$

where $u_{m,n}$, $v_{m,n}$ are the weighting coefficients of the approximation, and the $\phi(x-m, y-n)$ are the wavelet basis of the subspace V_0 at the fine resolution 0. Solving the optical flow problem requires the solution of the problem of finding the weighting coefficients of the approximation. Different types of images are used for the experiments. Experimental results showed that the approach was more accurate than the conventional methods.

Chapter 5

A New Resolution Mosaic Image Segmentation Algorithm

In this chapter the multiresolution analysis is utilised to address the problem of image segmentation for still images. A new algorithm based on the well-known Expectation Maximization (EM) algorithm [DLR97] is proposed. This new algorithm uses the wavelet transform and the multiresolution analysis to generate a resolution mosaic image for the segmentation.

As an example, the application of the segmentation of magnetic resonance images of the human brain is selected.

5.1 Motivation

The representation of an image in different resolutions enables easier extraction of local and global information than the original image. The use of multiresolution images enables to work with different type of information separated from each other. The lower the resolution of the image, the easier the extraction of global information. On the other hand, the higher the resolution of the image, the easier the extraction of local information and details.

If a scene has different regions, where some are more interesting than the others, then the need grows to have different resolutions for these regions. The interesting regions could be displayed in a higher resolution than the non-interesting regions. For example consider Fig. 5.1(a). It shows a general scene of a building in Cairo with a sky in the background and non-interesting objects such as the ground and the houses. Consider that the most interesting part of the building is in the centre of the image surrounded by buildings with less interest.

(a)

(b)

Figure 5.1: Ibn-Toloun mosque in Cairo. (a) Image in one resolution. (b) Image displayed in different resolutions for different regions.

In Fig. 5.1(b) the same image is displayed with different resolutions corresponding to the level of interest. The most interesting part is displayed in the original resolution, while the surrounding buildings are displayed in a one level lower resolution and all the rest of the image is displayed in the lowest resolution.

We can use the term *resolution mosaic image* to describe an image that consists of different parts with different resolution levels. The lower-resolution parts are simply higher-level approximations at the regions they represent. The resolution level used for a certain part should illustrate the level of the relevance of the information contained in that part. Thus, the structure of the mosaic in the image should illustrate the distribution of the relevance of the information contained in the image.

5.2 The Algorithm

5.2.1 Overview

The proposed algorithm is based on three assumptions that are reflected in the algorithm steps.

1. The algorithm assumes that after the segmentation the image is a *complete data* measured from a statistical field (Bayesian statistics). The pixel intensity is known *a priori*, usually called the *incomplete data*, and the field parameters are the *missing data*.

2. It assumes that the pixel's intensity cannot be the only feature for the segmentation. Other features, such as, (*i*) the pixel's location (the pixel may lie on an edge between segments or inside one segment), and (*ii*) the pixel's neighbourhood (the intensities of the neighbouring pixels must be taken into account by the segmentation process).

3. It assumes that the image subject to segmentation consists of relevant as well as non-relevant parts. For the non-relevant parts only low attention should be given for the local information.

To address the first assumption, a Gaussian mixture model is chosen to represent the statistical field and the EM algorithm is used to maximise the expectation of the missing data. For the second assumption, the multiresolution analysis is utilised so that the pixel and its neighbouring pixels are involved in the segmentation process. A resolution mosaic image is created, in which each pixel represents a block of pixels in the original image.

The resolution mosaic image is also used to address the third assumption. The parts of the image with relevant local information, such as edges, are processed with higher resolution, while the parts where no relevant local information is expected are processed with lower resolution.

The proposed algorithm lets the EM algorithm run on a resolution mosaic image, instead of running on the image in its original resolution.

The EM algorithm was introduced in Section 2.2.2 in *Statistical methods*. The resolution mosaic image is generated using many approximated version of the original image. To generate such approximations, any multiresolution technique can be used. However, for this work the wavelet transform is used. To create an image with different resolutions, a map is needed that describes the resolution level for each part of the image. This map can be called a *mosaic map*.

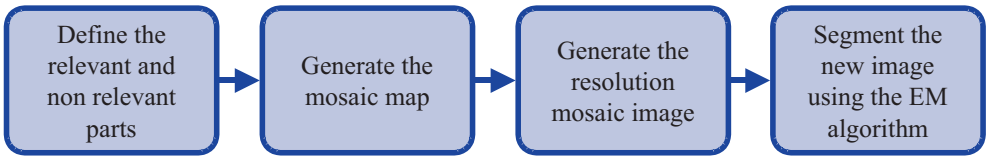
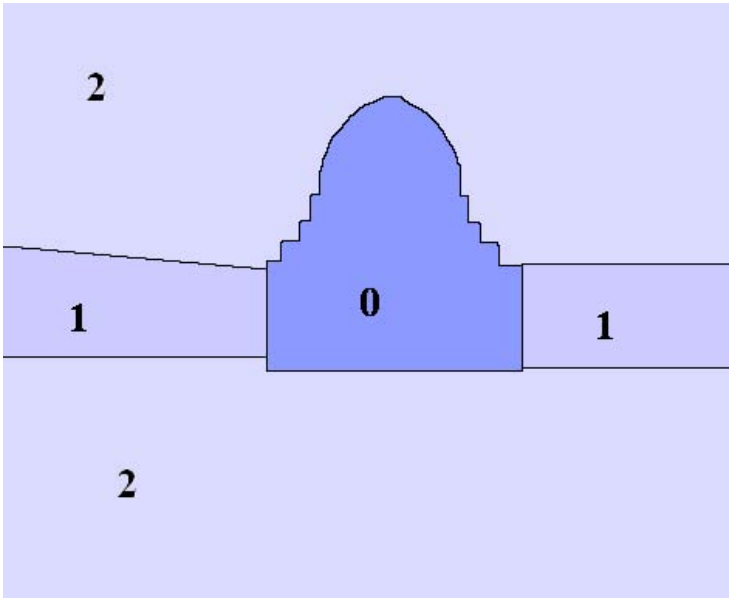


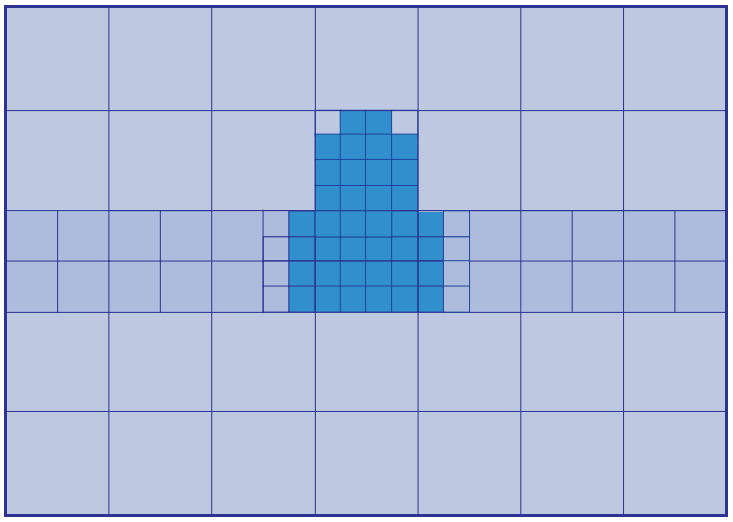
Figure 5.2: Steps of the new resolution mosaic image segmentation.

Fig. 5.2 shows a block diagram of the new algorithm consisting of four steps. First, there must be some sort of definition of high and less relevant regions. Based on this definition a mosaic map can be created which is used for establishing the resolution mosaic image. Finally, the EM algorithm is used for the segmentation.

Fig. 5.3 illustrates the generation of the resolution mosaic image of the image in Fig. 5.1(a). Fig. 5.3(a) shows an edge image that defines the parts with relevant local information (label value 0), parts with lower relevant information (label value 1), and parts with non-relevant local information (label value 2).



(a)



(b)

Figure 5.3: (a) Definition of relevance of the local information for the image in Fig. 5.1(a). (b) Corresponding mosaic map.

In Fig. 5.3(b) the corresponding mosaic map is shown, where the pixels are grouped in blocks based on the local information relevance level. The parts with high relevance level are left in the original resolution, while the pixels in the parts with intermediate relevance level are grouped in small blocks of size 2×2, and the pixels in the parts with the lowest relevance level are grouped in larger blocks of size 4×4. The corresponding resolution mosaic image is shown in Fig. 5.1(b).

5.2.2 Generating the Mosaic Map

The mosaic map image is used as a reference that defines the relevant and non-relevant parts of the image subject to segmentation. It is a label image, where the non-relevant parts are labelled with high numbers indicating a higher analysis level and a lower resolution. On the other hand, the relevant parts are labelled with low numbers indicating a lower analysis level and a higher resolution. The relevance of the information is determined by the intended application. For the applications tested in this work, the relevant information is the edge information. The pixels lying on edges need to be classified, without taking into account the information from the surrounding pixels. In contrast, for the pixels lying in regions away from edges, considering the information from the surrounding pixels can enhance the segmentation results.

By performing two levels of wavelet analysis on an image I, we obtain two successive approximation images A_1, A_2 and three detail images at each level, H_1, V_1, and D_1 at the first level, and H_2, V_2, and D_2 at the second level. The three detail images H_1, V_1, and D_1, are combined together to create a new image that contains only the edges of the original image, i.e., the horizontal, vertical, and diagonal edges that have been smoothed or blurred in the approximation image A_1. In Fig. 4.11 the information content of the approximation image A and the detail images H, V, and D were demonstrated.

In order to create such image the details H_1, V_1, and D_1 are converted to binary images by choosing suitable thresholds for each of them. Then these three binary images are combined in one image by performing the logical OR operation. This new image shall be called a *mask*. The pixel values can be either 1 or 0 depending on whether the corresponding pixel in the approximation image A_1 is generated from pixels lying on an edge or not. The same procedure is repeated on the detail subbands of the second level of the analysis giving a mask image for the approximation image A_2.

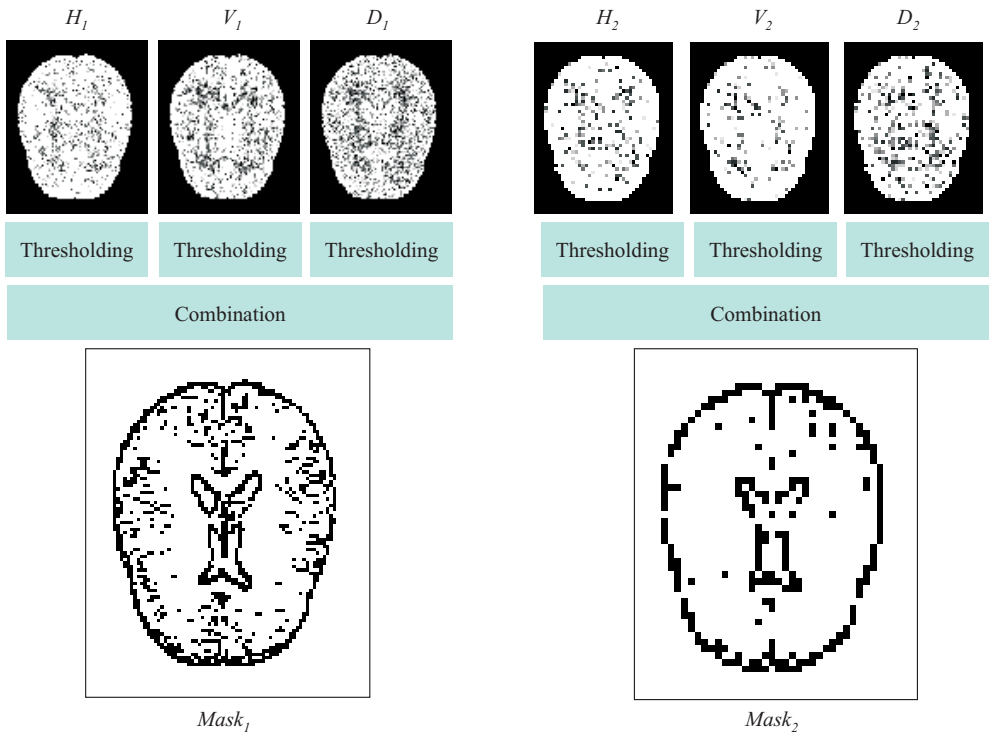


Figure 5.4: Creating two mask images from the corresponding detail images (MRI of Fig. 9.4).

An example for creating masks for an MRI is given Fig. 5.4. Note that the dimensions of the detail images of the second level H_2, V_2, and D_2 are half of the dimensions of the detail images of the first level H_1, V_1, and D_1. They are stretched for a better display, as the created mask images.

In the following step both mask images are used to generate a mosaic map image. An empty label image is created in the same size as the original image I, i.e., in the same resolution as the original image. Pixel positions in the original image are tested, if the corresponding pixels in the first level mask or in both level masks are lying on edges. Then the corresponding pixels in the mosaic map image are labelled with 0, since those pixels in the original image need to be processed in the original resolution. The corresponding pixels in the mosaic map image are labelled with 1, if the corresponding pixels in the second level mask but not in the first level mask are lying on edges. They are a bit away from edges and can be processed with a lower resolution. The pixels in the original image can be processed in small blocks in which each of four neighbouring pixels are considered to be one unit.

If the corresponding pixels in the first and second level masks are not lying on edges, then the corresponding pixels in the mosaic map image are labelled with the highest label value 2 to indicate that such pixels can be processed in the lowest resolution. Those pixels can be processed together with their neighbourhood pixels since they are located inside a homogeneous area away from a region's boundary or edges.

In the case of Fig. 5.4 the lowest resolution is the second level of the wavelet analysis, where each pixel represents a block of 4×4 pixels in the original resolution image. The generation of the mosaic map can be summarised as follows:

$$\mathrm{Map}(x) = \begin{cases} 0 & \mathrm{Mask}_1(x) = 1 \\ 1 & \mathrm{Mask}_1(x) = 0 \ \wedge \ \mathrm{Mask}_2(x) = 1 \\ 2 & \text{otherwise.} \end{cases} \tag{5.1}$$

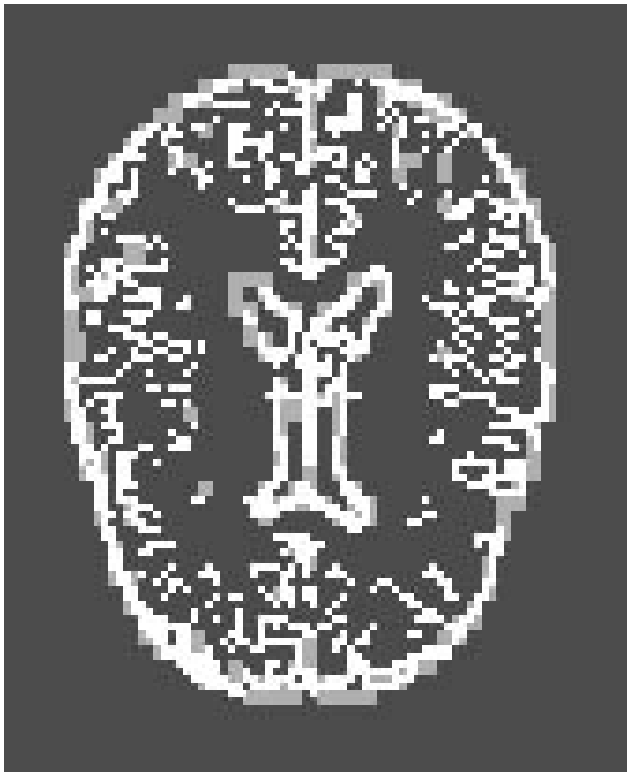


Figure 5.5: Generated mosaic map for an MRI.

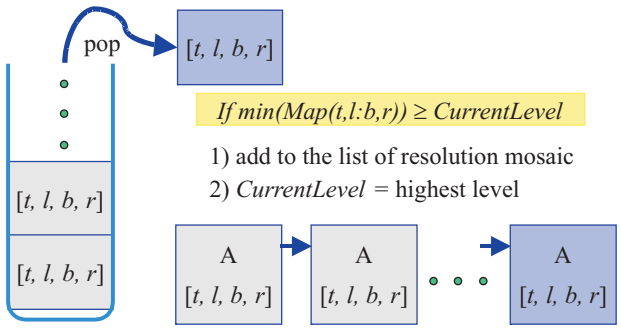
In Fig. 5.5 the generated mosaic map for an MRI contains three colours: the dark one is used for the lowest resolution level 2, the white colour for the highest resolution level (original resolution = 0), and the light grey colour for the intermediate resolution level 1.

5.2.3 Generating the Resolution Mosaic Image

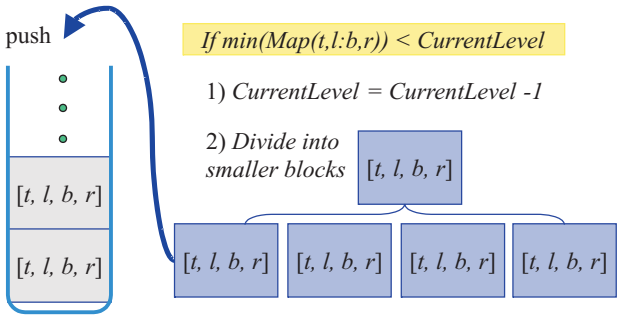
As soon as the mosaic map is ready, the next step is to generate the resolution mosaic image. The mosaic map is divided into non-overlapping blocks. The size of the blocks depends on the highest analysis level. The position and the dimension information of the blocks are saved in a stack such that the most top left block is on the top of the stack and the most bottom right block in the bottom of the stack. The position and dimension information is saved using the position values of the top t, left l, bottom b and right r pixels of each block relative to the image. This stack is called stack of the *non-processed blocks*.

The resolution mosaic image is then created by processing the blocks in the stack in a loop. For this, the value of the current analysis level is named as *CurrentLevel*. The block on the top of the stack is popped up and checked if it is possible to process it in the *CurrentLevel*. The label values of all pixels in the corresponding block of the map are checked if they are greater or equal to the *CurrentLevel*. If this condition is true, then the value of the corresponding approximation from the approximation image $A_{CurrentLevel}$ is added to a list together with the position and the dimension information of the block. Finally, the value of the *CurrentLevel* is reset to the highest analysis level for the next iteration of the loop. If the condition is not satisfied, *CurrentLevel* is set to the value of the next lower level. The block is divided into smaller blocks with dimensions corresponding to the new value of *CurrentLevel* and pushed back in the stack. This process is illustrated in Fig. 5.6. This way an image is created that consists of different blocks. Each of which has a resolution level depending on the label of the corresponding block in the resolution map image.

The new resolution mosaic image is archived using a simple list as a data structure. Each node of the list represents a block in the corresponding original image and saves the position and the dimensions of the block and its average grey level. A grey value can be either the original grey level of the image, if the block represents only one pixel, or it can be the approximation value computed by the wavelet analysis, if the block represents a block of pixels in a lower resolution.



(a) Pop and add operation



(b) Divide and push operation

Figure 5.6: Generating a resolution mosaic image from the corresponding resolution mosaic map. (a) Pop up operation and the replacement of an image block by the approximation of the current level. (b) Replacement of a block in the stack of non-processed blocks by four subblocks to be processed in a higher resolution (lower analysis level).

The nodes of the list can be redefined so that they save all the coefficients of the wavelet analysis, if the detail coefficients are needed for any further processing. The conventional data structure of the images (2D matrix) is not an efficient data structure for the proposed representation. It would mean to reconstruct an image with much redundant values. Moreover, in case that all coefficients have to be saved, then four images for the different coefficient subbands must be reconstructed.

5.2.4 Segmentation

The EM algorithm as a tool for segmentation works on the pixel values regardless of their spatial information. It is possible to run it directly on the grey levels saved on the list.

The use of the list simplifies the computation since the number of elements to be processed is reduced. In experiments the representation of the image in a list saved about 80% of the size of the image. Fig. 5.7 shows the steps of the proposed algorithm.

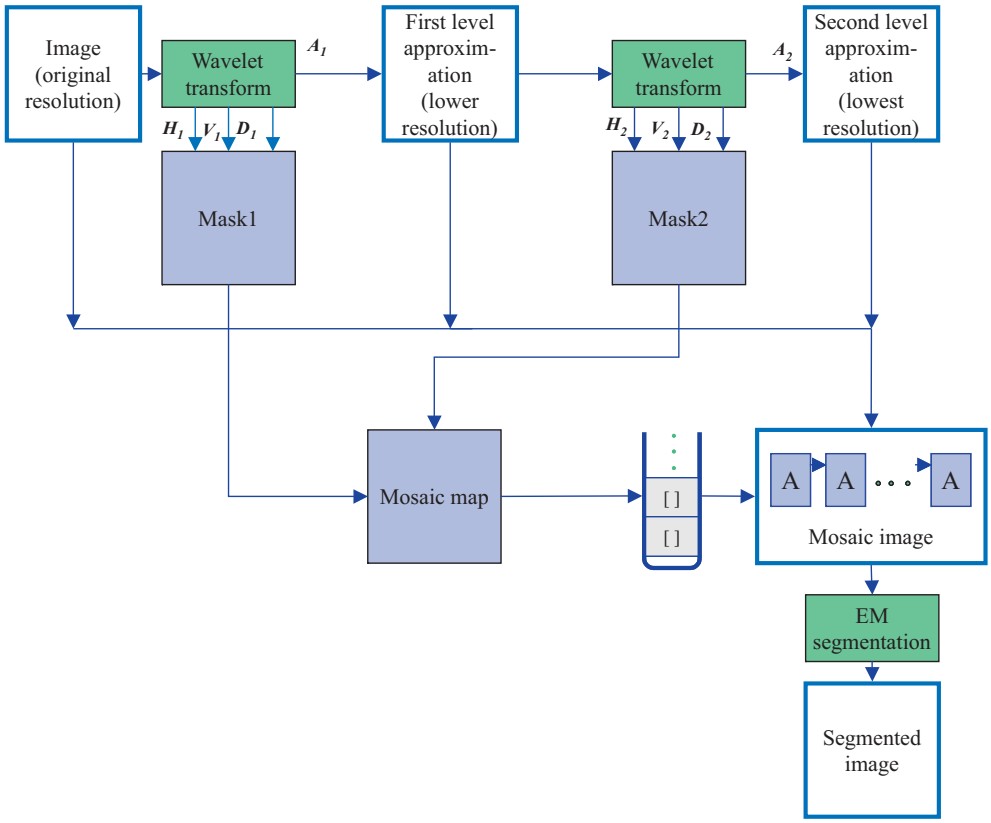


Figure 5.7: Block diagram of the resolution mosaic EM algorithm.

The grey level values in the list are given to the EM algorithm as a vector of data as well as the number of the expected classes (number of distributions in the mixture).

The result of the segmentation is a vector that represents the classification of the pixels in the corresponding input vector. The position and dimension information is then retrieved from the list and a new list of the classification is created. The classification values in this list can be easily written as a matrix, that represents the segmentation results in the conventional form. This form is needed for the evaluation and the comparison of the segmentation results with the conventional EM algorithm.

5.3 Discussion

The total time complexity of the new algorithm can be computed as the summation of the time complexity of the individual parts. The first part is to compute the lower resolution images. The Haar wavelet is used for this purpose, which is based on the simple operations (+, - and shifting). To compute the first level approximation and the details, each four pixels are processed only four times. To compute the second level approximation and details, each four pixels of the first level approximation are processed again four times. This yields a time complexity of:

$$\mathcal{O}(4\frac{N \times N}{4}) \; + \mathcal{O}(4\frac{N/2 \times N/2}{4}) \; \approx \; \mathcal{O}(N^2) \tag{5.2}$$

where $N \times N$ is the dimension of the image subject to segmentation. Generating the masks is an OR operation which needs an $\mathcal{O}(N^2)$, i.e., the time complexity of this part of the algorithm is $\mathcal{O}(N^2)$.

The two mask images are scanned once to compute the resolution mosaic map. The corresponding resolution mosaic image is constructed using the values of the pixels of the original image or the already computed values of the pixels of the first and second level approximations. In the best case, the new image is of size 1/16 of the size of the original image, while in the worst case its size is equal to the size of the original image.

The new image is a subject to the segmentation by the EM algorithm. The time complexity of the EM algorithm is linear with total length of the data [MCE04]. The input data for the EM algorithm are a list of values of the pixels, so its time complexity is $\mathcal{O}(N)$.

Thus the overall time complexity of the algorithm is $\mathcal{O}(N^2)$.

The application used to test the new algorithm is a medical magnetic resonance image (MRI) of the human brain. This application is chosen because results from other methods are available. In addition, other types of pictures have been tested, such as synthetic images and simulated magnetic resonance images.

The results of the segmentation are introduced in Chapter 9 compared to results of the segmentation by the standard EM algorithm introduced in Section 2.2.2.

Chapter 6

A New 3D Wavelet-based Video Segmentation Algorithm

In this chapter a novel segmentation algorithm for moving object detection is proposed. The algorithm is based on multiresolution analysis and the classical 3D wavelet transform. First, an introduction to the segmentation problem for traffic monitoring systems and a motivation to utilise the 3D wavelet analysis for solving this problem are given. After describing the algorithm various wavelets are investigated for traffic monitoring applications. Finally, masks created using a combination of different resolutions are used for improving the results. The chapter ends with a discussion of the performance of the new segmentation algorithm.

6.1 Motivation

The problem that we address here is the segmentation of image sequences in traffic monitoring that have been captured by a stationary camera.

The goal of a traffic monitoring system is robust extraction of traffic parameters. Traffic monitoring systems can be classified into vision-based systems [BMCM97, KM03, YYK03, ZK03a, ZK03b, BBRS04], non-vision based systems [ARS01, IVs03, Fas05], or hybrid systems [MBK$^+$05]. Vision-based systems are suitable for monitoring highways or intersections to analyse various traffic situations and scenarios because they have a powerful capability to extract more information than the non-vision based systems [YYK03]. It is possible to implement these systems as follows:

1. Only one stationary camera [BMCM97, CLK+00, YYK03, ZK03a, ZK03b, BBRS04, TCAA05], as investigated in this work.

2. More than one camera, each of which represents a different view of the same scene [KM03, MBK+05].

3. A moving camera, where a camera is attached in the front of a moving car [CJDC02].

4. Satellite images [ZN01].

All these devices and the analysis and processing algorithms are working together to build, at the end, an intelligent traffic control system, which enables dynamic traffic signal management and optimisation of traffic flow, especially in busy periods. In addition, dangerous situations can be reorganised and accidents prevented.

For traffic monitoring systems the first step is moving object extraction by image segmentation. As mentioned before, the aim of the conventional image segmentation is to divide the image into disjoint regions or classes, where all the pixels in a segment have some common characteristics and share a common meaning. In the case of traffic monitoring, the segmentation means a detection and isolation of the moving objects that take part in the actual active traffic situation. Because of different aspects that are related to the application, conventional algorithms for segmentation cannot be used. First, the input data are a sequence of images that represent the traffic parameters during a defined time period. Unlike the static images of the conventional problems, the image sequence has much relevant temporal information. A robust segmentation algorithm must utilise this information. Second, the aim of the segmentation algorithm should not be the separation of the homogeneous regions based on individual pixel's features, but extracting the active objects in the scene. It is not a trivial task to decide if an object is moving or not. In a traffic flow there can be a stop-and-go or a short time parking situation where an object seems to be non-active. The same is true for the background that may have moving and non-moving parts.

The proposed algorithm has the advantage of considering the relevant spatial as well as temporal information of the movement. A movement in time sequence images is a 3-dimensional change, two spatial dimensions and the time. Under this assumption, the detection of the moving objects is the answer to the question where and when there is a change in the local and temporal information.

The algorithm benefits from the multiresolution characteristics of the wavelet analysis. The analysis of an image sequence is done in different resolution levels. This speeds up the processing and improves the performance of the segmentation. In recent years multiresolution representation of images has got significant attention. But it has not been widely used for segmentation of time sequence images.

6.2 The Algorithm

6.2.1 Overview

Two types of results are expected from the algorithm: the detection of the moving objects, or simply the extraction of regions of interest (ROI), and the extraction of the active traffic area. Moving object detection is done by extracting a mask from a group of frames. This mask represents the ROI in this group. The number of frames in a group depends on the level of the wavelet analysis. For finding the active traffic area it is sufficient to extract only a single mask for the complete data set.

The proposed algorithm consists of three parts as shown in Fig. 6.1. In the first part, the 3D wavelet analysis is used for moving object detection in the image sequence and it can be considered as a primary segmentation step. The second part is a conventional procedure to improve the segmentation and to provide binary masks for the ROI. The final part is a projection of the created masks onto the original images to extract the ROI from the original image sequence.

6.2.2 Detection of Motion

The first and main part of the proposed algorithm is the analysis of the input image sequence by the 3D wavelet transform. Three levels of analysis have been investigated. To apply the 3D analysis the input sequence is divided into group of frames. For the first level of the analysis only two frames are analysed at once, for the second or the third level four or eight, respectively. As already shown in Fig. 4.12, the results of the 3D wavelet transform are eight subbands, but only the subbands D^4 and D^7 are used for the further processing. In general, the detail subbands have low intensity values. The subband D^4 shows relative high intensity values where events have occurred in time, while the subband D^7 shows events in all three dimensions.

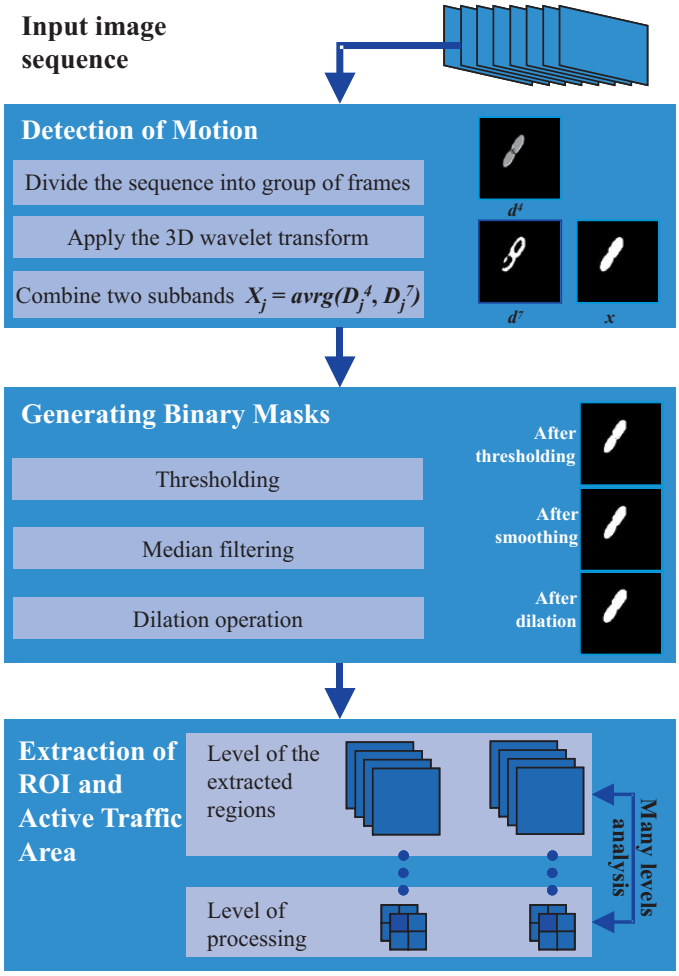


Figure 6.1: Block diagram of the 3D wavelet-based algorithm.

As explained in Section 4.4, D^7 represents a great part of the motion information in changes of the spatial and temporal domains. However, its results show only the borders of the moving objects. It has been found that combining D^7 with other subbands improves the results. Comparing different subbands, the subband D^4 has been found to give the best results. The subband D^4 represents the change between the approximations of the successive frames. It shows the area where the movement occurs clearly. However, any changes in the pixel's intensity between the processed frames also appear around the moving region, i.e., the results are very noisy. A simple average is used for the combination with the subband D^7.

The output of this step is a sequence of images with low intensities, which in most cases are in the range between 0 and 80 in the grey level scale. The images have their highest intensity values where the movements are occurring and values near zero otherwise. Each group of frames of the input sequence is represented by only one image in the output sequence. The output of this step can be considered as a primary segmentation that needs enhancements. Fig. 6.2 shows an input image and the corresponding primary segmentation resulting from this step.

(a) (b)

Figure 6.2: (a) One input image of an image sequence. (b) Output of the primary segmentation.

6.2.3 Creating Binary Masks

The aim of the second part of the algorithm is to create a binary mask. This is done by thresholding the output of the wavelet analysis, followed by a smoothing step using the median filter and a region-growing step.

Different simple techniques for thresholding were tested. The best results were obtained by the following technique, which can be summarised in three points.

1. Compute the cumulative histogram: The cumulative histogram of the grey level images consists of two parts. The first fast ascending part represents the background. The second slow ascending part represents the foreground with high grey levels. The point, where the ascent rate changes, is the border point that differentiates between background and foreground. This is the point of interest to be found by the technique.

Fig. 6.3(a) shows the histogram of the image shown in Fig. 6.2(b). The blue line in Fig. 6.3(b) is the corresponding cumulative histogram. In both figures the border point is illustrated by vertical lines. The cumulative histogram shall be modified in such a way that a new (concave) curve is created with a maximum at the border point.

2. Compute the corresponding concave curve: The concave curve can be computed by dot-multiplication of the cumulative histogram and a descending function (green line in Fig. 6.3(b)) which can be defined as following:

   (a) It should have a descending rate slower than the ascending rate of the first part of the background. The result of the multiplication is still an ascending curve for this part.

   (b) It should have a descending rate faster than the ascending rate of the second part of the foreground. The result of the multiplication will be a descending curve for the second part.

   The resulting concave curve is shown in Fig. 6.3(c).

3. Select the position of the maximum value as a threshold: The grey level value of the maximum of the concave curve is taken as a threshold between background and foreground. The image after thresholding is shown in Fig. 6.3(d).

The median filter is used here to remove the noise that may arise from small movements in the background, e.g., the movements of tree leaves or changes in lighting conditions.

Median filtering is a neighbourhood operation, in which the value of a pixel in the output image is determined by selecting the middle value after sorting the values of the neighbourhood of the corresponding input pixel.

It has been found that applying the smoothing step on the binary image after thresholding gives better results than applying the thresholding after the smoothing. Fig. 6.4(a) shows the binary image in Fig. 6.3(d) after smoothing by the median filter.

The last step in this part is a region-growing step using a dilation operation. This operation adds pixels to the boundaries of objects in an image. The number of pixels added depends on the size and shape of the structuring element, which is used to process the image.

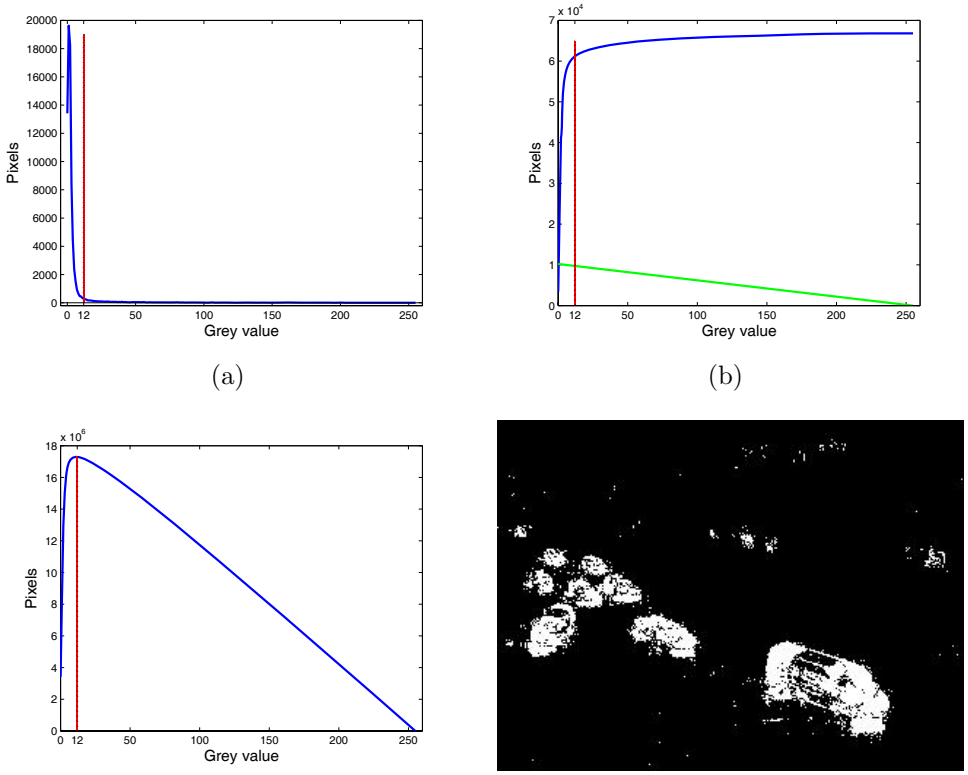
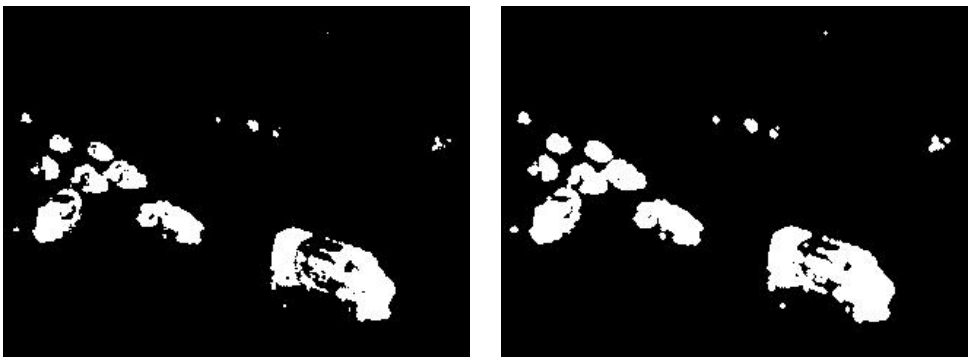


Figure 6.3: (a) Histogram of the image in Fig. 6.2(b). (b) Cumulative histogram. (c) Cumulative histogram after the multiplication by a descending curve. (d) Binary image after thresholding.

Similar to the median filter the dilation operation is done on a neighbourhood of pixels. The value of the output pixel is the maximum value of all the pixels in the input pixel's neighbourhood. In a binary image, the output pixel is set to 1 if any of the input pixels has the value 1. The structuring element defines the neighbourhood of the pixel of interest. The centre pixel of the structuring element, called the origin, identifies the pixel of interest. The structuring element can be of any size and shape [Cas96, GW05] but is typically much smaller than the processed image. This way the dilation connects any two subregions that may be separated by one or two pixels. Furthermore, the dilation is able to fill the holes inside the mask. These black holes inside the extracted regions in the mask are due to the low pixel values in the detail subbands D^4 and D^7 of the inside parts of the moving objects.

The chosen sizes of the median filter and the structuring element for the dilation operator depend on the data set. Typically, for the median filter the used size was 5×5 and for the dilation operator a diamond structure of radius between 3 and 5 was used.

This part of the algorithm can be considered as a segmentation enhancement step. Fig. 6.4(b) shows the resulting mask obtained from the first level analysis for the image in Fig. 6.2(a).



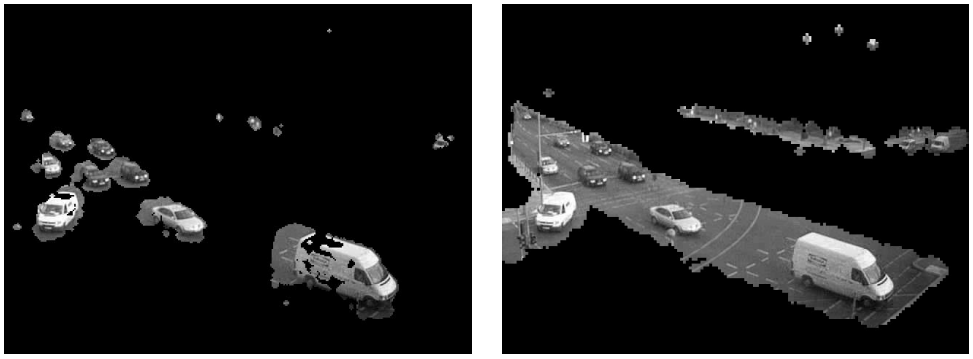
(a) (b)

Figure 6.4: (a) Median filtering applied to the image of Fig. 6.2(a). (b) After dilation operation.

6.2.4 Extraction of Interesting Regions

Each of the masks generated in the last part represents the ROI in a corresponding group of input images. To extract the ROI, each group of input images and the corresponding mask image are processed using the logical operator AND. The result of the extraction using the first level of analysis of the image in Fig. 6.2(a) is shown in Fig. 6.5(a).

Because the masks are in lower resolutions than the original images, a projection is required. Practically, this projection is done pixel-wise during the logical operation. Each pixel from the mask is operated with a cube of pixels representing the corresponding group of sub-blocks of the input images.



(a) (b)

Figure 6.5: (a) Extraction of the ROI using first analysis level. (b) Extraction of the active traffic area using second level.

It is a key issue that the monitoring system should be adaptable enough to find solutions for problems, which require little or no a priori knowledge of the analysed scene. Adaptive traffic monitoring systems should be able to face changing conditions, such as changing light levels and changing analysis goals. Automatic active traffic area detection plays an important role for that aim [KZK03]. To extract the active traffic area of a scene a mask from the whole image sequence or from images of a relatively long period has to be updated. Using the proposed algorithm this goal can be achieved simply by cumulating the extracted regions from the different masks. This has to be done until no more relevant changes are observed. Because each extracted mask represents the motion of several successive images, only a few masks need to be added to get the complete extraction of the active area. In Fig. 6.5(b) an extraction of active traffic area is given using the second level of analysis.

In the case of a busy active traffic situation other combination methods of the masks can be used. For example, averaging the masks can lead to better results due to suppression of the unwanted small motions derived from moving objects in the background.

To evaluate the algorithm for ROI extraction many data sets were used and evaluated manually by a human operator. For the extraction of active traffic areas, the results are evaluated using manually segmented images. The results and the evaluation are presented in Chapter 9.

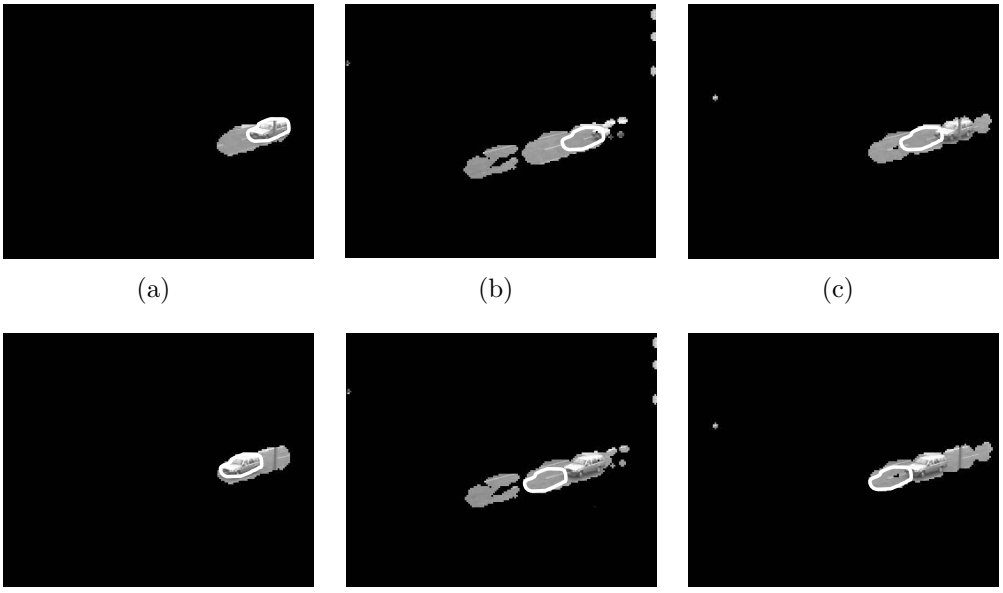
6.3 Using Different Mother Wavelets

To determine the influence of different mother wavelets to the ROI extraction, three wavelets from the Daubechies family are tested and evaluated against each other, namely Haar, DB4, and DB8.

As introduced in Section 4.2.4 the lengths of the filters associated with the wavelet functions Haar, DB4, and DB8 are 2, 8, and 16, respectively. The filtering operation is done in the mean of a window that slides over the input data points with dyadic shifting and computes an output coefficient. This window has a size equal to a j-multiple of the size of the associated filter, where j is the analysis level. If the window size is equal to the shifting step, then the analysis is done without overlapping and the number of the resulting coefficients is equal to the number of the required shifting to cover all the input data points. Otherwise, the analysis is done with overlapping, i.e., the data points are used several times to compute different coefficients. Only for the Haar wavelet the size of the output signal is half the size of the input signal and the analysis is done without overlapping. The analysis of the wavelets DB4 and DB8 is done in an overlapping manner and output signals are longer than the half of the size of the input signal, for wavelet DB4 three points larger and for wavelet DB8 seven points. This discussion is valid for images and videos. It concludes a "too" early and wide detection of events. For example, in case of images, the edge pixels are used to compute coefficients away from the corresponding positions in the output subbands. The same is true for videos, where the movements are predicted temporally and detected in a "too" big ROI.

Because of the asymmetry of the wavelet functions DB4 and DB8 the detection of a ROI has one more drawback, which is the shifting of the ROI. The extracted ROI have the form of the moving objects, but they appear shifted left and so they miss some of the right parts of the moving objects.

Simple and complex scenes were used in the tests and it has been found that the results become worse as the order of the wavelet increases. The results of the Haar wavelet are the only acceptable ones among the three tested wavelets. This fact is illustrated in Fig. 6.6. The extracted regions are shifted and show different areas, which, in some cases, do not contain any of the moving objects. As shown in Figs. 6.6(e) and 6.6(f), the extracted regions correspond to a too early detection of the moving objects if they move from right to left and a too late detection if they move in the opposite direction. Because of these inconvenient results, all further testing and improvements of the algorithm are based on the Haar wavelet.



(a) (b) (c)

(d) (e) (f)

Figure 6.6: Extraction of the region of interest in two successive frames by the first analysis level using (a) and (d) Haar. (b) and (e) DB4. (c) and (f) DB4.

6.4 Using Interresolution Masks

Up to this point, three different results were computed from three different resolution levels that are independent of each other. As an enhancement of the 3D wavelet-based algorithm, the resulting masks from the different resolutions are combined to create an interresolution masks.

If the algorithm is modified by logical operations as illustrated in Fig. 6.7 the quality of the ROI or the active traffic area can be increased.

Five different combination methods are presented. The simplest combinations assume that the ROI in the interresolution mask contains either all parts of the ROI in all the three levels, or only the parts that are common in all ROI in the three levels. The first assumption is achieved easily by using the logical OR operator, while the second is done by the logical AND operator. These combination methods can be expressed as:

$$\mathrm{ROI}_{comb1} = (\mathrm{ROI}_{level1} \vee \mathrm{ROI}_{level2} \vee \mathrm{ROI}_{level3}) \tag{6.1}$$

$$\mathrm{ROI}_{comb2} = (\mathrm{ROI}_{level1} \wedge \mathrm{ROI}_{level2} \wedge \mathrm{ROI}_{level3}) \tag{6.2}$$

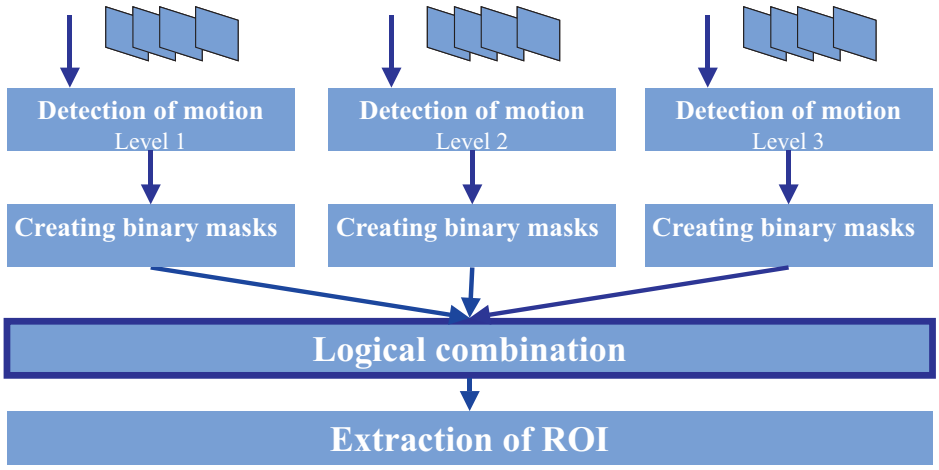


Figure 6.7: Block diagram of the 3D wavelet-based algorithm with the modification of the interresolution masks.

The third combination method assigns a pixel to a ROI if the corresponding pixels belong to a ROI in the first level and any of the other two levels. This method gives the results of the first analysis level the key role. It assumes that the results of the first level are nearly true and need a justification from at least one of the other two results to support the decision.

In the fourth and fifth combination methods the key role is shifted to the second and third analysis level. These combination methods can be expressed as:

$$\mathrm{ROI}_{comb3} = (\mathrm{ROI}_{level1} \wedge (\mathrm{ROI}_{level2} \vee \mathrm{ROI}_{level3})) \tag{6.3}$$

$$\mathrm{ROI}_{comb4} = (\mathrm{ROI}_{level2} \wedge (\mathrm{ROI}_{level1} \vee \mathrm{ROI}_{level3})) \tag{6.4}$$

$$\mathrm{ROI}_{comb5} = (\mathrm{ROI}_{level3} \wedge (\mathrm{ROI}_{level1} \vee \mathrm{ROI}_{level2})) \tag{6.5}$$

All combination methods have the same complexity. In all cases the algorithm must be computed for the three analysis levels. The complexity is not much increased, because the size of the input sequence decreases very fast against the increase of the analysis level. Computing an interresolution mask means to perform an overhead of less than 15% of the size of the original sequence.

6.5 Discussion

In this discussion we deal with the analytical evaluation of the introduced algorithm. The evaluation concerns the measure of an asymptotic upper bound for the usage of computational resources regarding the size of the input data, i.e., the complexity of the algorithm against the input data. The input of the algorithm is a variable length sequence of images of different sizes. The size of an image is important for the analysis whereas the length of the sequence is not a deciding parameter.

The 3D wavelet-based algorithm is divided into three parts. Each part is evaluated alone. Then an overall evaluation will be performed. The evaluation is expressed in terms of the number of operations multiplied the complexity of each type of the operation.

The main task of the first part is the application of the 3D wavelet transform which computes the primary segmentation. Each group of frames is analysed independently. Each group is divided into small cubes of dimension $k \times k \times k$, where k depends on the level of analysis j, i.e., $k = 2^j,\ j = 0, 1, 2, \ldots$. The type of the operation done on each cube is the wavelet transform with complexity $\mathcal{O}(k)$. For k frames of size $N \times M$ the complexity of the 3D wavelet transform is of order:

$$\left(\frac{k}{k} \times \frac{N}{k} \times \frac{M}{k}\right) \mathcal{O}(k^3) = \mathcal{O}\left(k^3 \times \frac{N}{k} \times \frac{M}{k}\right)$$

For $N > M$ one may consider k^3 to be constant, since $k \ll N$ and it is fixed for a chosen analysis level. Thus, as an upper bound the complexity of this part of the algorithm can be assumed to be $\mathcal{O}(N^2)$.

From the eight subbands resulting from the transform, only two subbands are then subject to the next *average* operation, which in general has the complexity of $\mathcal{O}(N)$ for 1D data. In our case it has the complexity of $\mathcal{O}((\frac{N}{2})^2)$, since the dimensions of each subband is at most $\frac{N}{2} \times \frac{N}{2}$. Thus the complexity of the first part of the algorithm is:

$$\mathcal{O}(N^2) + \mathcal{O}\left((\frac{N}{2})^2\right) = \mathcal{O}(N^2)$$

The inputs of the second part are sequences of grey level images of size $\frac{N}{2} \times \frac{N}{2}$. The first operation, the thresholding, has three steps. *First,* computing the histogram, which has the complexity of $\mathcal{O}(N^2)$. *Second,* selecting the maximum value. This operation has a constant complexity $C = 256$. *Third,* setting the values of the pixels either to 1 or 0, which has the complexity of $\mathcal{O}(N^2)$ for images. Then the complexity of the thresholding is:

$$\mathcal{O}(N^2) + C + \mathcal{O}(N^2) \;=\; \mathcal{O}(N^2)$$

After this operation the algorithm works on binary images, where each pixel is represented by only one bit. This simplifies the further processing. However, this simplification will not be considered in the current evaluation. The second operation of this part is the application of the median filter. It has the complexity of $\mathcal{O}(N^2)$ since it is based on selecting a window of size $v \times v = w$ for each pixel in the image and then sorting the elements in that window with operation complexity $\mathcal{O}(w \log w)$. But, since $w \ll N$ (in our implementation of the algorithm v is usually set to 5, so $w = 25$), the complexity of the sorting operation can be considered as a constant. However, it must be repeated for all pixels in the image.

The last operation in this part is the morphological operation *dilation*, which works completely as the median filtering and so it has the same complexity.

Then the whole complexity of this part of the algorithm is:

$$\mathcal{O}(N^2) + \mathcal{O}(N^2) + \mathcal{O}(N^2) \;=\; \mathcal{O}(N^2)$$

Finally, the binary masks created in the last step, are projected to the original images using a logical operator. This operation is done one time for each pixel, which implies a complexity of $\mathcal{O}(N^2)$.

The overall complexity of the algorithm is $\mathcal{O}(N^2)$ where N^2 is the size of the images in the sequence.

To measure the computation complexity of the wavelet analysis, we limit the analysis for the orthonormal wavelets and follow the analysis introduced in [LOPR97]. Using the algorithm of Mallat [Mal89] the size of the processed signal at each level of the analysis is halved. Assuming a signal S of size N. For the first level of analysis the complete signal is processed. For each further level the size of the signal is halved, i.e., the first approximation A_1 has $N/2$ values, the second approximation A_2 has $N/4$ values, and so on. Assuming the length of the associated low-pass filter as p, then filtering the signal needs $p \times N$ multiplications, and so:

$$\begin{aligned}\left(N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + ...\right) \times p &= p \times \sum_{i=0}^{\infty} \frac{N}{2^i} \\ &= p \times 2N \\ &\approx \mathcal{O}(N)\end{aligned}$$

considering p as a constant depending on the mother wavelet used.

The Haar wavelet transform needs no filtering, it needs only $+$, $-$, and shifting operations, which are very hardware friendly and need no digital signal processing blocks.

Chapter 7

A New Resolution Mosaic Video Segmentation Algorithm

Based on the results obtained from the previous algorithm, a new segmentation algorithm for moving object detection in video surveillance applications is proposed. It is based on the wavelet packet analysis and the resolution mosaic representation of images. The chapter begins by describing the drawbacks of the previous algorithm. Overcoming these drawbacks is the motivation for the new resolution mosaic segmentation algorithm proposed in this chapter.

7.1 Motivation

In the previous algorithm, the input sequence of images is analysed for a certain number of levels in the spatial domain as well as in the temporal domain. The data are transformed with equal spatial and temporal resolutions. Although the obtained results are better than those of the conventional algorithms in the literature, they can be enhanced if a good temporal resolution is used with different spatial resolution. The drawbacks of the previous algorithm are:

1. The strong dependence between the resolution of the spatial and the temporal analyses is not in all cases helpful for the detection of objects. Low spatial resolution helps to suppress the irrelevant motion interference in the background of the scene. However, low temporal resolution means to miss fast motions. In other words, due to difference in the spatial and temporal information, a freedom to choose different spatial than the temporal resolutions is needed.

2. A high spatial resolution may be required only for the detection of small objects, which can be seen in the far view of the scene. The background and the close-up view of the scene, in contrast, need to be analysed in low spatial resolution. Thus the scene needs to be analysed in various spatial resolutions.

To address the first demand the temporal dimension of the input sequence should be analysed independently on the spatial dimensions. Hence, the images are first transformed spatially by the 2D wavelet transform for arbitrary n levels. Then all coefficients of the last n^{th} spatial level are transformed by a 1D wavelet transform temporally for arbitrary m levels. This analysis agrees with the 3D wavelet packet analysis [FR07].

To address the second demand, the scene should be analysed in various spatial resolutions. The far views in the scene are analysed in high resolutions. In such far views the objects appear smaller and move slowly. The close-up views are analysed in low resolutions, because the objects usually appear bigger and move faster than in the other parts of the scene. The background can be analysed in very low resolution, since no relevant information is expected from it. Therefore, it is proposed to transform each image in the sequence into a resolution mosaic using different levels of the 2D wavelet transform. Fig. 7.1 shows a simple description of the suggested resolution mosaic.

7.2 The Algorithm

7.2.1 Overview

The algorithm shown in Fig. 7.4 consists of the same three parts as the 3D wavelet-based segmentation algorithm. The resolution mosaic and the 3D wavelet packet analysis are performed instead of the 3D wavelet transform for motion detection. As before, the results can be considered to be a primary segmentation. The second and the third part are left unchanged. The second part is used to enhance the results of the segmentation. Finally, the third part is used to extract the ROI from the original input image sequence. In this chapter the focus is on the motion detection.

The result expected from the algorithm is the extraction of regions of interest. As before, a mask is created which represents the ROI for a group of frames.

7.2.2 Generating the Mosaic Map

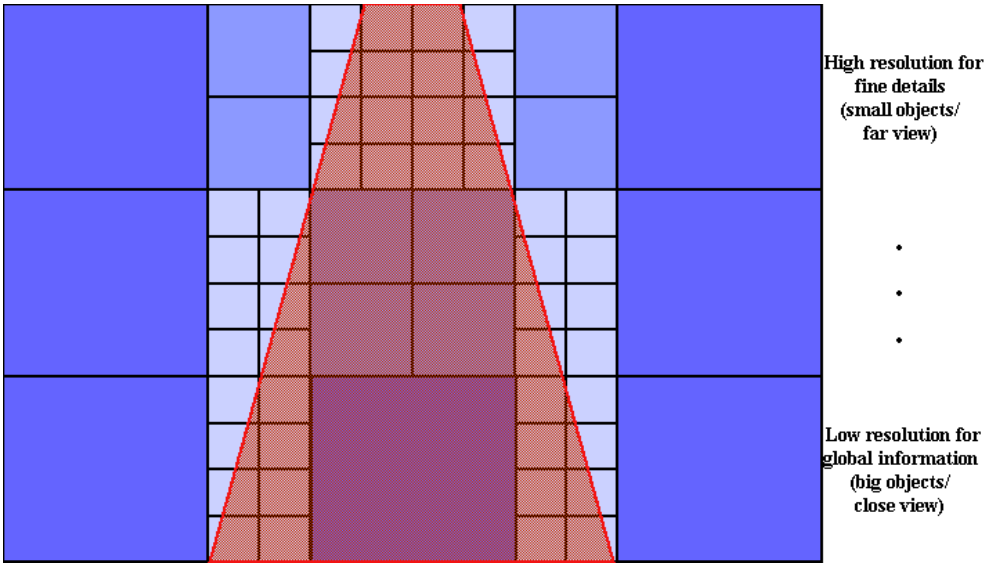
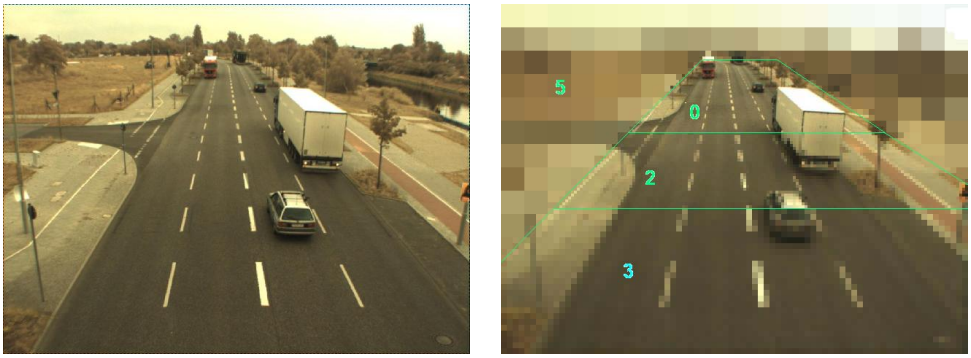


Figure 7.1: Suggestion to mosaic a scene in different resolutions.

Processing the image in different resolutions helps to adapt the detection algorithm to the size of the objects and the speed of the movement. In addition, it helps to prevent irrelevant movements from taking part in the detection. Each scene may need its own mosaic map. A mosaic map as defined in Section 5.2.2 is a label image, where the non-relevant parts are labelled with high numbers, indicating high analysis levels and low resolutions. On the other hand, the relevant parts are labelled with low numbers, indicating low analysis levels and high resolutions.

Generally, up to six resolutions are used for almost all the processed scenes: five wavelet analysis levels and the original resolution. The lowest resolution is normally used to represent the background. Processing the background in a low resolution may be better than to exclude it at all, since for some applications, like security monitoring by a human operator, the background is important for conventional examination.

Fig. 7.2 shows an example of a scene that is composed of four different resolutions. The image in Fig. 7.2(a) has the original resolution and the image in Fig. 7.2(b) is shown in the resolution mosaic. The background is represented by the fifth level approximation, i.e., each block of 32×32 pixels is represented by only one value.



(a) (b)

Figure 7.2: Example of the resolution mosaic of a scene.

The foreground, the active traffic area, is divided into three parts. The close view is represented by the third level approximation, the far view is left in the original resolution and the part in between is represented by the second level approximation.

The generation of the mosaic map is done either manually by a human operator, or automatically.

A human operator was asked to define the background and the foreground parts of the scenes. Then the foreground was divided into regions based on the change in size and speed of the moving objects and the amount of the traffic activity. For example, the close views with fast and big moving objects should belong to one region, while the far views with slow and small moving objects should belong to another region. The different regions were given labels that represent the relevance of the local information.

The resulted active traffic areas obtained from the application of the previous 3D wavelet-based algorithm were used as a priori information for the human operator. They were very important to figure out the borders between the background and the foreground. Since, some margin parts of the background are covered by the moving objects. Therefore, there were no expected results of the type active traffic area. They would be in all cases non-distinguishable results from that obtained by the previous algorithm.

In order to have an automatic generation of a mosaic map, the estimated masks for the active traffic area are used that were estimated using the second combination method of the interresolution masks corresponding to Eq. 6.2. Only a few frames are processed.

After updating the mask by the last group of frames $mask_{new}$, it is compared to the current mask $mask_{current}$ before it's updating. If the change between the two masks is smaller than a predefined threshold, then the process stops. Otherwise, the process continues for the next group of frames. The estimated mask is used as a simple mosaic map, where the background is given the lowest resolution level, and the active traffic area is given a suitable resolution level. This level should be a compromise between the resolution used for the close views and the resolution used for the far views in the previous manually created maps.

7.2.3 Detection of Motion

For each processed scene only one resolution map is created manually. If this map is available the incoming image sequence can be processed.

The next step is to generate a resolution mosaic image. To do this the mosaic map is divided into non-overlapping regions based on the given pixel labels. Each new observed frame is divided into corresponding regions. Each region is then analysed by the 2D wavelet transform. The number of levels is equal to the value of the pixel labels in the corresponding region of the mosaic map. If the original regions are placed by the resulting approximation coefficients of the last analysis level then we have a mosaic of regions in different spatial resolutions. This shell be called mosaic image of the approximation subband. Four mosaic images are created for each frame, namely a mosaic image for each subband of the last analysis level. Thus, we have an image for the approximation coefficients, A, and the horizontal, H, vertical, V, and diagonal, D, detail coefficients. This process is continued until all mosaic images of a group of frames are computed.

The next step is to perform temporal analysis for the third dimension for the available group of frames. The spatially corresponding blocks (scalar values) from the mosaic images of the four subbands are grouped in vectors. These vectors represent temporal arrays which contain the 2D wavelet coefficients of the same position (row and column) of a (successive) group of frames. The vectors are then transformed by the 1D wavelet transform.

Outputs of this step are the temporal approximation and the detail coefficients for each input vector. Thus, the results of this step are eight subbands AA, AD, HA, HD, VA, VD, DA, and DD.

This process is illustrated in Fig. 7.3. It is a combination of the spatial 2D wavelet transform and the temporal 1D transform.

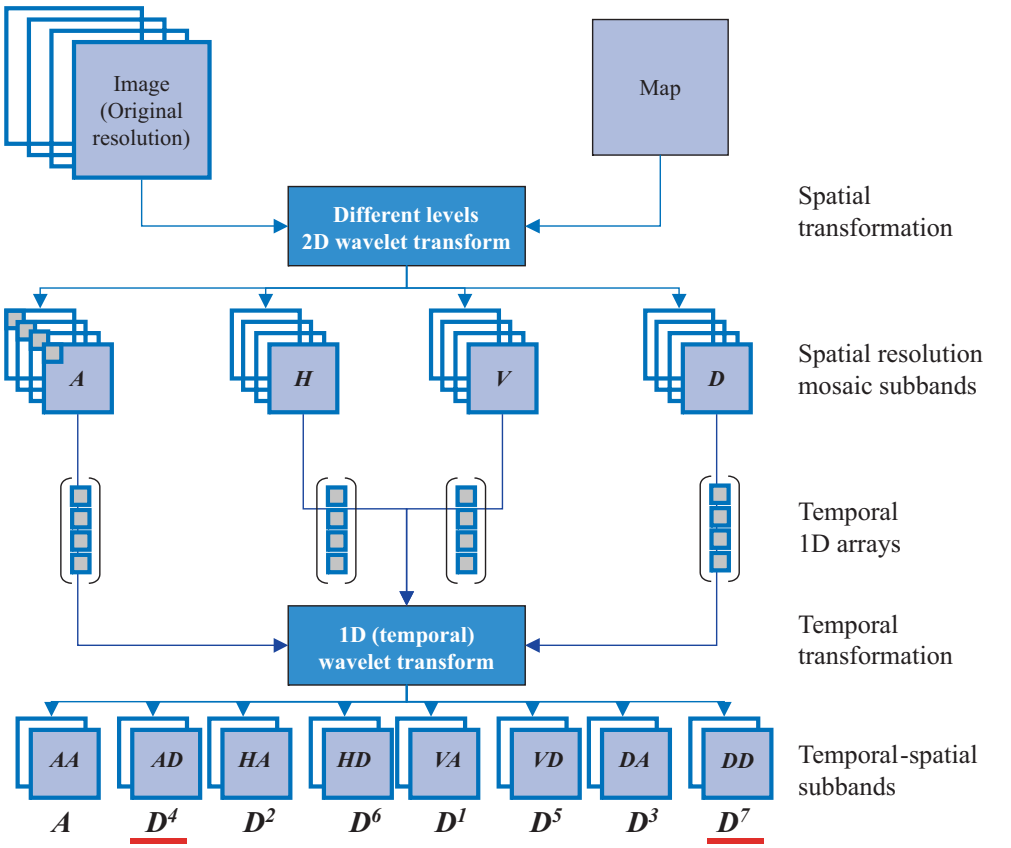


Figure 7.3: Block diagram for the 3D wavelet packet analysis (2D spatial resolution mosaic + 1D temporal) for motion detection.

It gives eight output subbands that correspond to the output subbands of the conventional 3D wavelet transform. However, it differs from the conventional process in two points. First, the numbers of the spatial analysis levels and the temporal analysis levels are different. Second, the number of the spatial analysis levels is defined for each individual image region based on the information content.

The data structure used to represent the resolution mosaic images is similar to that proposed in Section 5.2.3. The approximation and detail coefficients of the last analysis level as well as the block position and dimension information are saved in a list for each frame.

Finally, similar to the 3D algorithm, the AD and DD coefficients, which correspond to D^4 and D^7 in the conventional 3D transform, are combined by averaging to give a primary segmentation. The new coefficients are ready for the following steps in form of lists instead of the conventional 2D matrix form.

7.2.4 Creating Masks and Extracting Interesting Regions

The second part of the algorithm is almost the same as the corresponding part of the 3D wavelet-based segmentation algorithm. It has been adapted to deal with the used data structure.

The thresholding can be performed on the subband in its list form, since it concerns no spatial information. But the smoothing and the dilation are neighbourhood operations, which take into account the spatial information between different blocks. Therefore, the list is converted to the conventional 2D matrix form before these operations were performed. A block diagram of the algorithm is shown in Fig. 7.4.

In Chapter 9 the results of the algorithm are evaluated against the results of the other algorithms.

7.3 Discussion

The automatic creation of the mosaic map has been found to be not effective in traffic monitoring applications. If in the beginning of the observation a part of the actual active traffic area is not active for a long time interval then a "too early" conversion between the $mask_{new}$ and the $mask_{current}$ is achieved. A "too small" estimated active traffic area is yielded and a part of the actual active traffic area is enclosed in the estimated background. Therefore, only the manual method is used to generate the mosaic maps.

It has been found in the application of the algorithm that combining the resolution mosaic with high resolution temporal analysis leads to good results. Therefore, the input sequences are usually subject to one level analysis only by the 1D wavelet transform in the time domain.

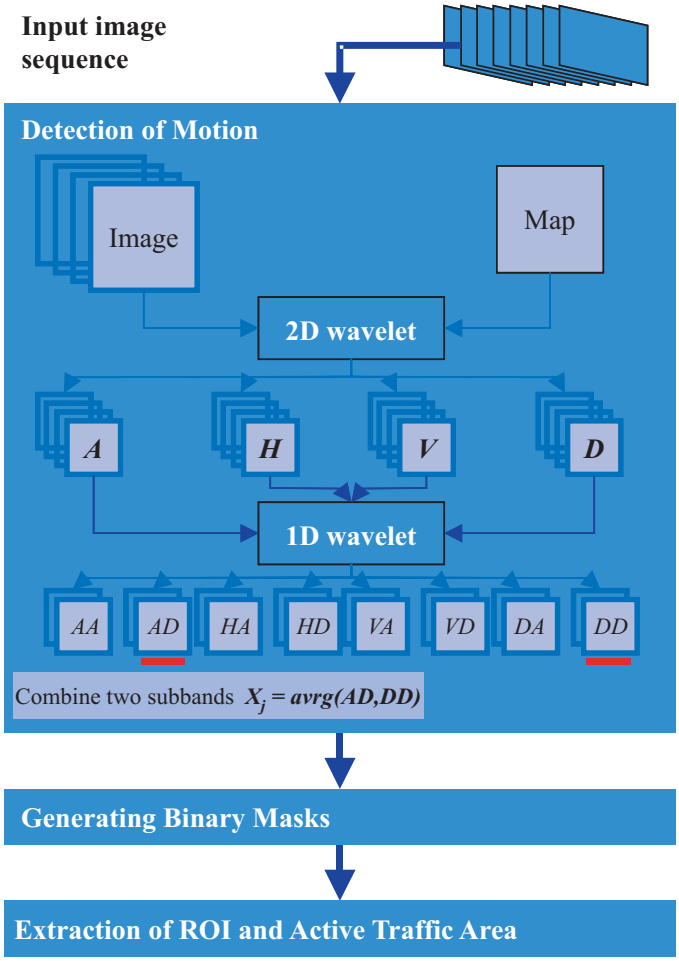


Figure 7.4: Block diagram of the segmentation algorithm based on resolution mosaic and the wavelet transform.

The experiments with different data sets show much better results than that obtained by the 3D wavelet-based algorithm described Section 6.2.

Determination of the resolution mosaic for each image in the input sequence may add computational overhead in comparison with the previous algorithm. However, the time complexity remains in the order of $\mathcal{O}(N^2)$. It has been shown in Section 5.3 that the time complexity for the generation of the resolution mosaic images is $\mathcal{O}(N^2)$. The second and the third part of the algorithm have the same complexity of $\mathcal{O}(N^2)$, as shown in Section 6.5.

Chapter 8

A Concept of Hardware Implementation

The first part of the 3D wavelet-based algorithm used for the primary segmentation was implemented partially in hardware. The purpose for using hardware was to utilise the inherent parallelism property of the wavelet analysis and its computational simplicity. The aims and the benefits of the implementation are discussed in the following section. The rest of the chapter is divided in two main parts. First, a general concept for the hardware implementation of the 3D wavelet transform is introduced. Second, a specific design of hardware for the motion detection is described.

8.1 Motivation

In most cases, a software solution for signal processing tasks is preferred which is executable on commercial off-the-shelf hardware. This is due to the availability of development tools which are matured and enable a simple implementation for even complex algorithms. In addition to this, modifications of the implementation are possible at any time. However, the algorithm will be executed on the CPU in a sequential order. Faster execution time can be achieved with higher clock frequency only. Added to that, the algorithm does not use the whole CPU and unused parts consume power.

Hardware solutions have the advantage to utilise the complete available resources. Unnecessary power will not be consumed. Hardware implementation is reasonable for such algorithms where simple operations as calculating sums and differences are required. Hardware solutions have the disadvantage of a high development effort and are less flexible. The implemented hardware cannot be modified any more.

The 3D wavelet-based algorithm is based on simple techniques such as Haar wavelet transform and frame differencing. It needs mainly simple operations as addition and subtraction. Furthermore it has high parallelism properties. All of that motivates to propose a hardware implementation for it. The hardware implementation was performed for the first part of the 3D wavelet-based algorithm which delivers a primary segmentation for moving object detection. In the implementation attention has to be given to the image acquisition to achieve real time processing. Up to this point, the design can be considered as a *smart camera*, because in addition to image capturing, it can extract information from images without the need for an external processing unit. The remaining parts of the algorithm can run on a client PC, which can be connected to the hardware implementation using a TCP/IP-based network. Thus, the hardware can be a part of an integrated vision system. Such a system includes sensor devices (camera), special hardware for real time processing, and other powerful computing units.

Fig. 8.1 shows an integrated system, which includes optical sensors, a small special hardware board, and powerful computing units such as servers and personal computers. The components are connected using Ethernet.

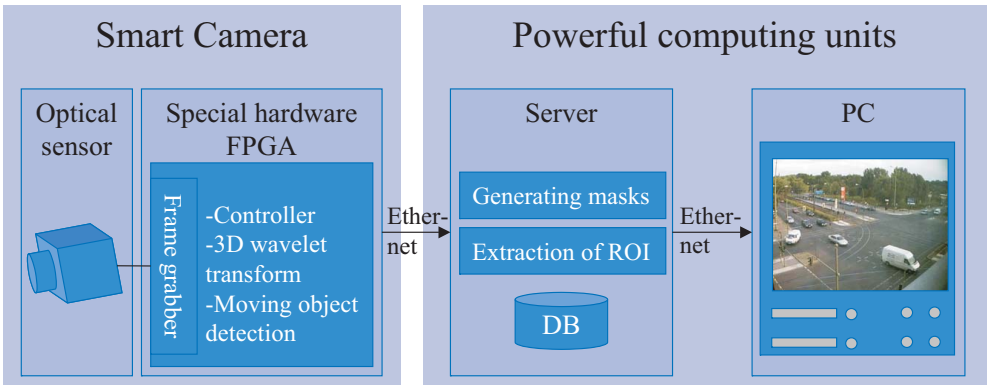


Figure 8.1: Components of an integrated system for moving object detection for traffic monitoring.

8.2 Implementation of the 3D Wavelet Transform

In this section a general concept for hardware implementation for the 3D wavelet transform is introduced [SAWM08]. The implementation is proposed for the Haar wavelet, since it is used in the proposed algorithms.

The implementation of the Haar wavelet transform is very simple. A single component, that is able to do very basic operations such as addition, subtraction, and shifting, is sufficient to implement the first level of the Haar wavelet transform. Two data points of the input signal are the inputs of such a component. The approximation (addition and right shifting) and the detail (subtraction and right shifting) are the outputs. This component could be called a *basic component*. Fig. 8.2 shows an illustration of it.

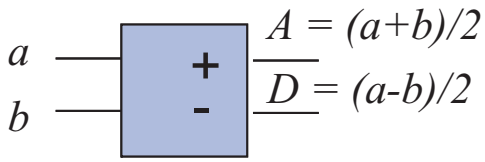


Figure 8.2: Basic hardware component for the Haar wavelet transform.

For the 2D Haar wavelet transform four basic components have to be used. The components are organised in two layers: two components in the input layer and two components in the second layer. The inputs are four data points, which represent 2×2 pixels of the input image, while the outputs are the approximation (A) and the horizontal (H), vertical (V), and diagonal (D) details.

For the 3D Haar wavelet transform 12 basic components are required. They are organized in three layers. The inputs are eight data points representing a spatial-temporal cube. The input cube represents a part of two consecutive images. Eight coefficients are the outputs, namely, the conventional eight subbands of the 3D transform.

An overview of the hardware design for the 3D case is given Fig. 8.3. It shows how the dimensions are built recursively, which gives the possibility to extend it as desired.

For n-dimension analysis the number of the input elements rises to 2^n. Hence the number of the basic components rises to $2^{(n-1)}$ for each layer in a depth of n layers. An n^{th} dimensional component is built using two $(n-1)^{\text{th}}$-dimensional components and an output layer consists of $2^{(n-1)}$ basic components.

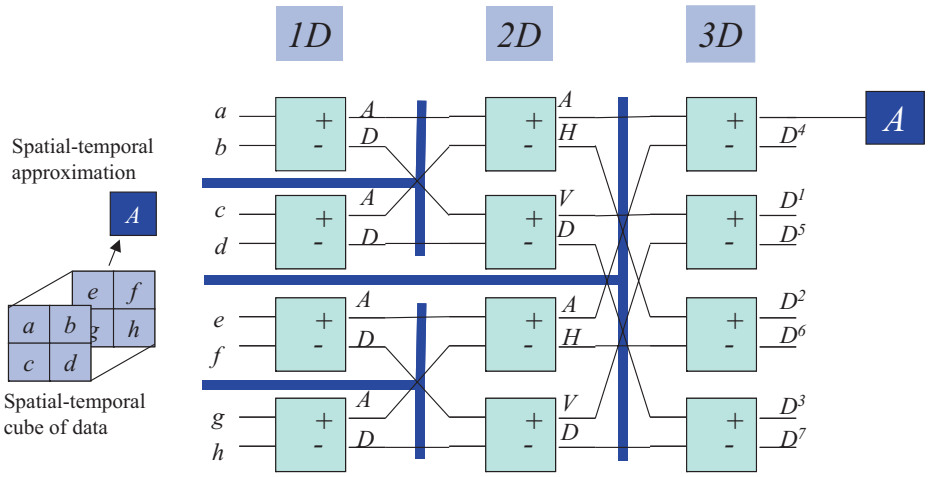


Figure 8.3: Hardware design for the 3D Haar wavelet transform.

For a multilevel analysis, the design can be done recursively. For the $(n)^{\text{th}}$ level, three components of the $(n-1)^{\text{th}}$ level are required, two components as input layer and one component for the output layer. Fig. 8.4 shows the design for two levels of a 3D analysis.

The grade of parallelism is limited by the available memory, which permits parallel access to the stored information. It depends on the storage format for the input data. For best performance, a video should be saved in a cube-wise format. Each cubic data is saved at one memory address. A cube of dimensions $2 \times 2 \times 2$ and 8 bits per pixel requires 64 bits memory space and permits only a one level 3D transform. A cube of dimensions $8 \times 8 \times 8$ pixels permits up to three levels but requires a 512-bit parallel memory access. Fig. 8.5 shows a cube storing concept for a 3D wavelet transform of a $16 \times 16 \times 2$ image sequence. The video data input has to be written at a certain sequence in the positions of the memory, marked by arrows in Fig. 8.5. In a hardware implementation the computation of this write position is very easy. Only simple address slice operations are necessary, which need neither time nor any hardware resources. On the other hand, only one memory read cycle gives an access to all 8 pixel values for a cube dimension of $2 \times 2 \times 2$. All computations of this cube can be done in parallel.

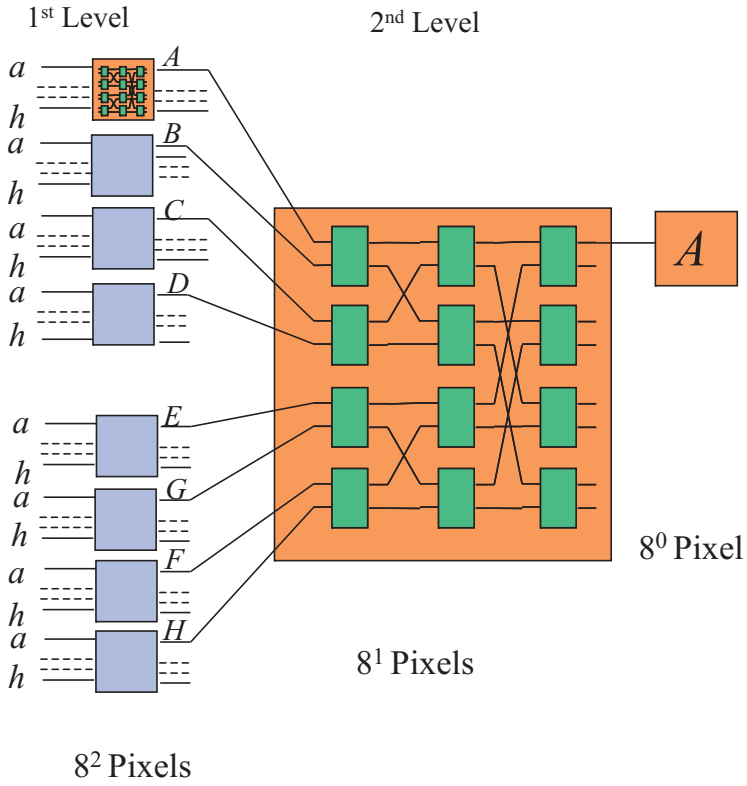


Figure 8.4: Hardware design for two levels 3D Haar wavelet transform.

8.3 Hardware-based Motion Detection

The purpose of this section is to introduce an implementation of the first part of the 3D wavelet-based segmentation algorithm. The goal is to have a hardware implementation for the image acquisition and the primary segmentation for motion detection as fast and as parallel as possible. Another goal is to be able to embed this implementation within an image processing system. Such a system integrates this hardware design with other high performance computing units allowing the running of the rest of the algorithm and any other requirements that may be asked by the end user.

An embedded system is a computer system that can be attached to electronic devices to do special tasks. It consists of hardware and software parts, which build one functional unit together. The software is responsible for the control of the hardware, input/output, and for the communication with other devices.

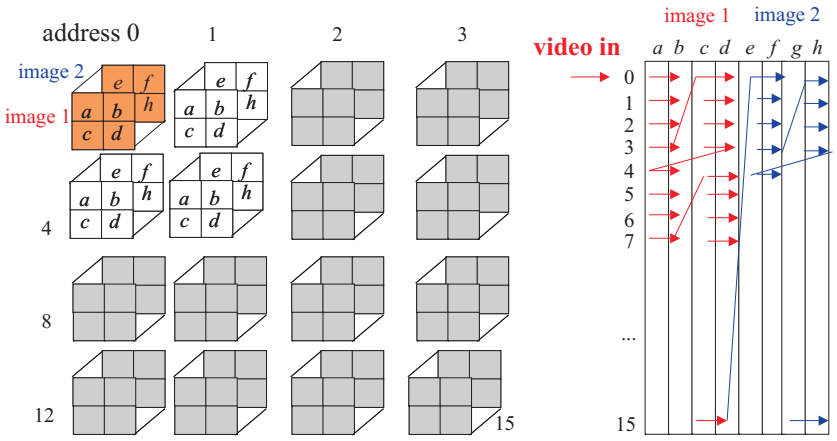


Figure 8.5: Cube storing concept for 3D wavelet transform.

A widely used example of the embedded systems is an FPGA (*field programmable gate array*). It is a low cost hardware platform containing a number of *configurable logic blocks* (CLB), *memory blocks* (BlockRAM), and programmable interconnects. Logic blocks can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex computational functions such as decoding or simple mathematical functions. In most FPGA, the included memory elements may be simple flip-flops or more complicated blocks of memory.

In Fig. 8.6 an example is given of a camera with embedded FPGA board. The FPGA board is configurable over a serial interface to allow the user to activate the automatic white balancing and to control temperature and the colour space. Moreover, it makes the captured data available on the camera link interface.

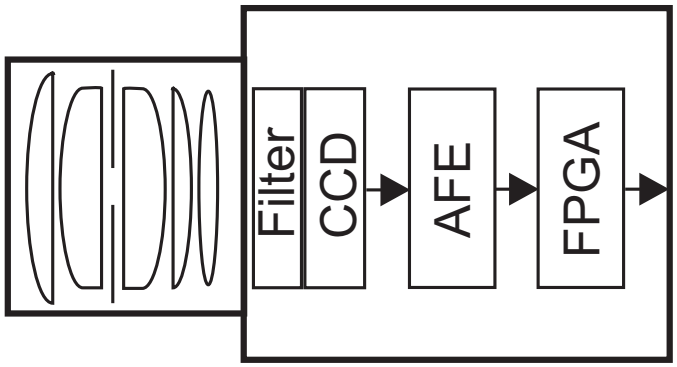


Figure 8.6: Embedded FPGA in a camera design.

In contrast to an ASIC solution (Application Specific Integrated Circuits), an FPGA can be reconfigured. Among other advantages, this reduces the development expenditure. The design can be simulated in software and tested on an FPGA board. Errors can be fixed in short time without producing a new chip. While ASIC development expenses are relatively high, FPGA expenses increase with the number of production pieces. Therefore, implementation on FPGA is suitable for experimental purposes and low production.

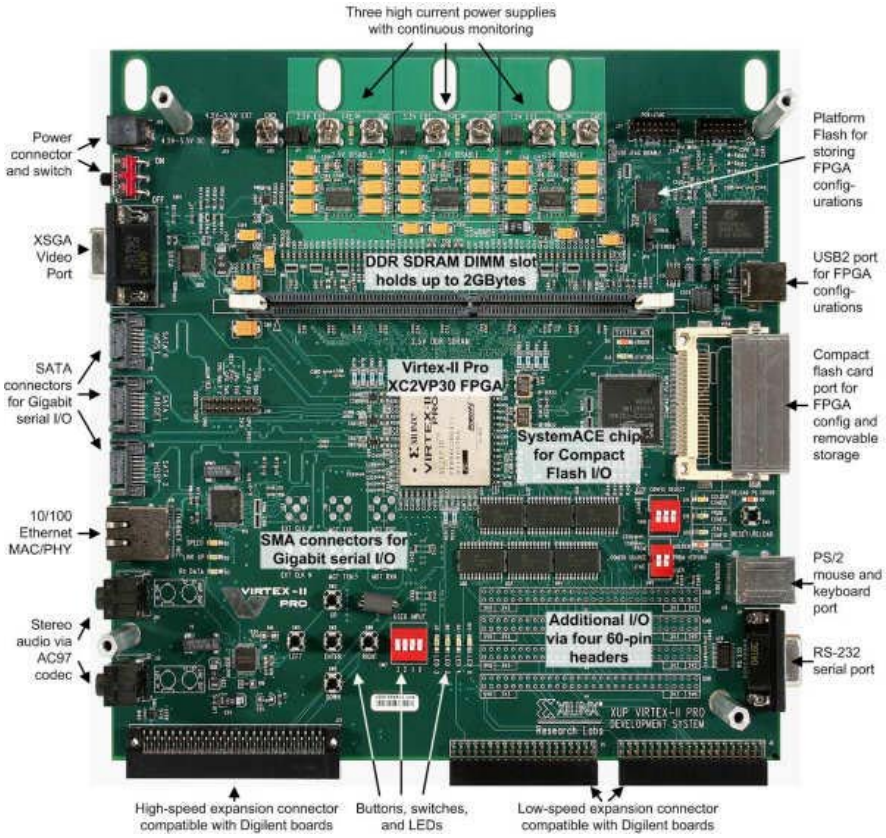


Figure 8.7: FPGA board XUP Virtex II (XC2VP30).

For the implementation proposed in this work an FPGA board XUP Virtex II Pro from the company *XILINX* [Xil05] is used. It has, among many other resources, two *embedded PowerPC-CPU* cores (PPC405).

One CPU is used to run an operating system for communicating with other devices via Ethernet. The other CPU configures some of the attached resources such as two SRAM memory modules, which are added to the board to store intermediate results. Both CPU cores are realized by an embedded Linux operating system. Fig. 8.7 shows the used board for the implementation.

The XUP board is not ready to be connected directly to a digital camera for image acquisition. Therefore, an extension *Video Decoder Board* (VDEC1) is used. This board receives analog TV signals like PAL or NTSC. The signal is converted into digital form using the analog digital converter ADV7183B on the board. Both, PAL and NTSC, send the image information in two parts. They transfer half of the image in a frame of all odd horizontal lines, followed by the second half in a frame of all even horizontal lines. Differences between PAL and NTSC exist in the image resolution and in the transfer rate. So PAL has a resolution of 720×576 pixels and a transfer rate of 25 images per second, while NTSC has a lower resolution of 720×486 but a higher transfer rate of 30 images per second. The digital camera used gives output in either PAL or NTSC. The analog digital converter is configured through a C program that runs on a PowerPC core on the FPGA.

The smallest input of the 3D wavelet transform is a cube of pixels. In our case the cube consists of two images. In this context we call them odd image and even image. The 3D wavelet transform cannot be implemented as shown in Fig. 8.3 because the frame grabber gives only half of the image every time segment. Instead, the processing is done in three 1D steps and needs four time segments to produce the first results of the 3D transform.

As shown in Fig. 8.8(a), various modules realise the processing of the acquired images. The first one *Clk_Sync* synchronises the frame grabber and the system clock. The following module computes the 1D wavelet transform. In the first time segment (T1) the first half (all odd horizontal lines) of the odd image is acquired from the camera and then transformed by 1D wavelet transform in the x-dimension. The results have to be stored temporarily until the second half of the odd image is available for the 1D transform in the y-dimension. Usually the internal memory of the FPGA is not large enough to store an image. Therefore, an external memory is used. A memory card, say SRAM-1, from the company *DIGITAL* is connected to one of the extension-ports of the FPGA board. The memory capacity used is 512-KB SRAM. A PAL-image is of size 720×576 pixels. Each pixel is represented by 2 bytes. So a frame (half an image) needs approximately 405 KB.

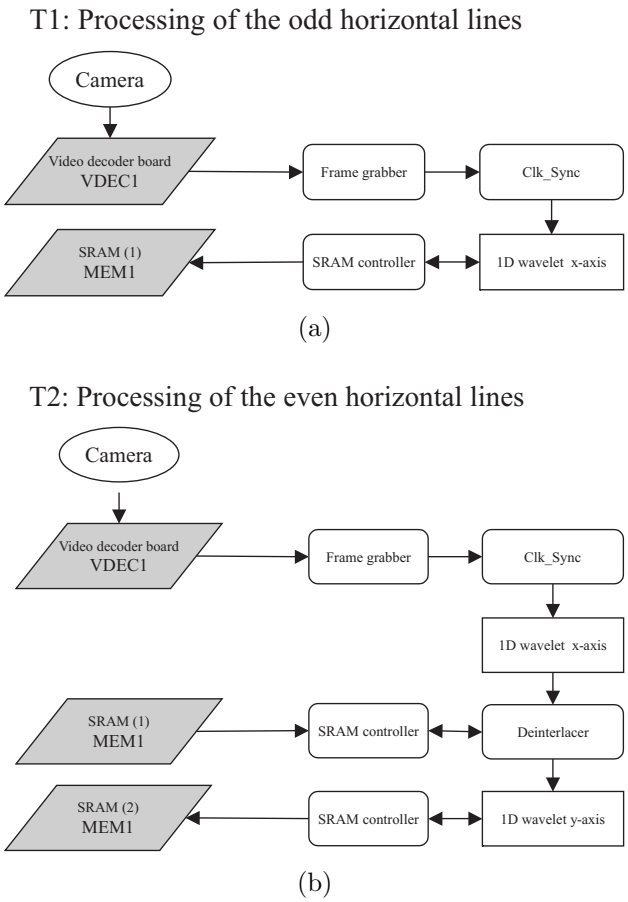


Figure 8.8: Illustration of the sequence of processing. (a) Time segment T1. (b) Time segment T2.

As shown in Fig. 8.8(b), the second half (all even horizontal lines) of the odd image is available in time segment (T2). It is processed first by a 1D wavelet transform in the x-dimension. Then, the already transformed first half is called from the memory SRAM-1. The *Deinterlacer* module merges both parts of the 1D wavelet transform results. Both halves are then transformed using 1D wavelet transform in the y-dimension. This completes a 2D wavelet transform. The results must be stored in another memory, say SRAM-2, in order to be used later when the results of the 2D transform of the even image is ready. For this purpose, a memory capacity of 1-MB SRAM is sufficient to store a full image (two frames).

In the next time segment (T3), the procedure done in (T1) is repeated, but this time on the first half of the even image.

The 3D wavelet transform of two consecutive images is completed in the time segment (T4). Fig. 8.9 shows that the processing in this time segment is similar to that in time segment (T2). However, in contrast to (T2), the results of the 2D wavelet transform are directly transferred to the *Time-Deinterlacer*. At the same time the results from the 2D wavelet transform of the odd image are called from the memory SRAM-2. Both images are then transformed in the third dimension by the wavelet transform. The results are stored in DDR-RAM on the XUP board using the Multi Port Memory Controller2 (MPMC2) [Xil06] from *XILINX*.

A web server running on Linux transfers the results to a network client PC for any further image processing. The web application has direct memory access to the wavelet transform results and transfers the data via TCP/IP to the network client. Because of the shared memory access managed by MPMC2 both processes work independently. The TCP/IP protocol is independent of the operating system and valid for different types of host PC.

8.4 Discussion

All acquired images are transformed with 25 fps (PAL) or 30 fps (NTSC). The bottleneck in this system is the 100-Mbit Ethernet interface transferring only six complete results per second. Therefore, the application supports the delivery of partial results with full frame rates, namely the subbands A, D^4 and D^7.

With the subbands A and D^4 it is possible to reconstruct the approximation of both input images used in the current processing. The images in the approximation subband A represent the original sequence in lower spatial and temporal resolution. The original temporal resolution can be restored by performing a 1D inverse wavelet transform in the temporal dimension. This is done with the help of the subband images D^4. The two successive images of the group of frames used in the processing can be restored by adding and subtracting the corresponding image (AAA) from the subband A and the corresponding image (AAD) from the subband D^4, respectively, but in a lower spatial resolution (AA_1 and AA_2).

If only grey level images are needed, the system can deliver all results with a full PAL/NTSC frame rate. The designed system is called *Smart Camera*, because it does image acquisition as well as processing for special application.

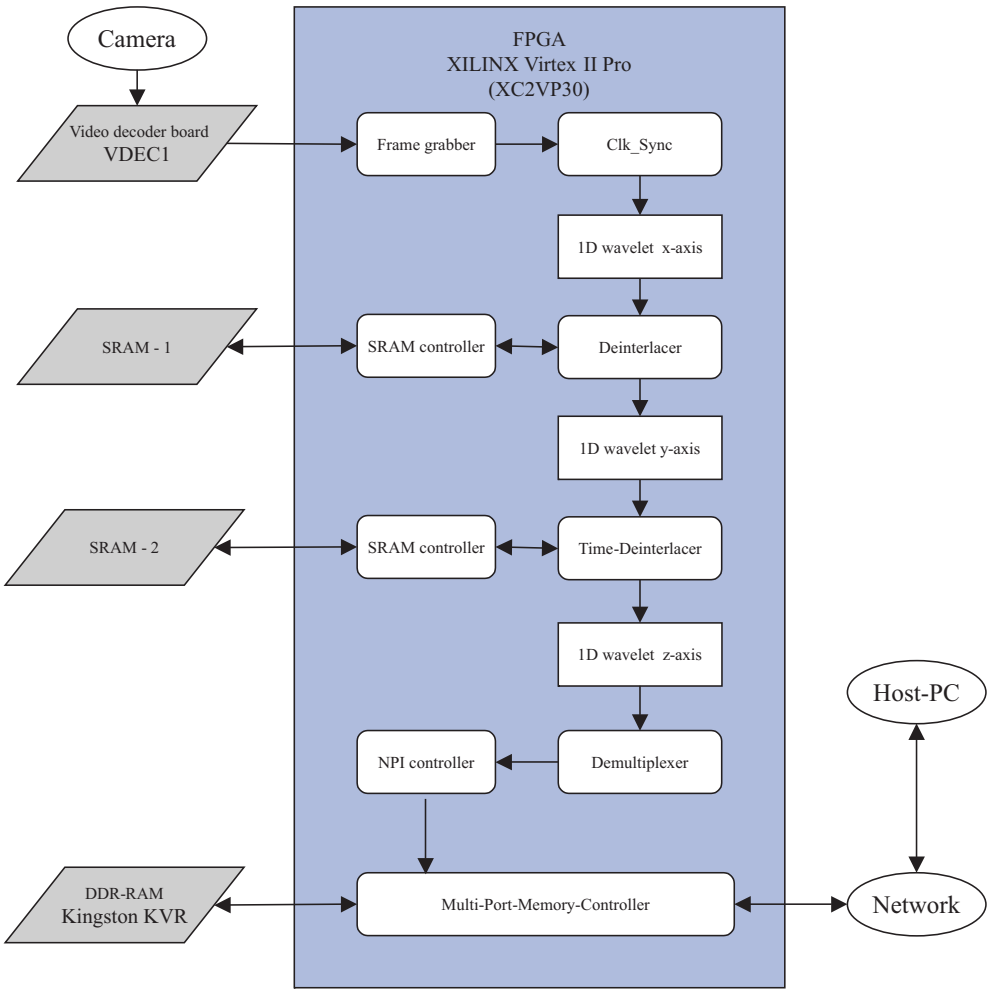


Figure 8.9: Implementation of the 3D wavelet transform on FPGA.

To overcome the limitation of the used board, another Virtex-II platform, the Alpha Data ADM-XP can be used. This development board has more independent memory banks: four banks DDR SSRAM and two banks DDR SDRAM. It is possible to implement the 3D wavelet transform and to deliver the results via 1-Gbit Ethernet without loss. A special extension board is designed with a camera link interface and a 1-Gbit Ethernet interface for image acquisition of 1000×1000 colour pixel with a frame rate of 30 fps. Due to the fact that this design is still under development, the results based on this board are beyond the scope of this dissertation.

Chapter 9

Results and Discussion

In this chapter the results are discussed which could be achieved for image segmentation and moving object detection using the proposed algorithms. The algorithm for image segmentation described in Chapter 5 is applied for the segmentation of synthetic and medical images. The segmentation algorithms for video surveillance application described in Chapters 6 and 7 are tested for moving object detection in traffic monitoring.

For the purpose of comparison, results of the expectation maximization algorithm and the 2D wavelet-based background estimation algorithm, which are introduced in Chapter 2, are also presented.

Preceding the description of the results and the discussion, in the first section the test data sets are described. The second section contains the methods used to evaluate the performance of the algorithms.

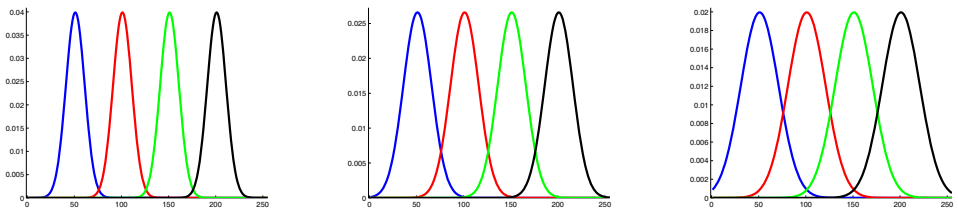
9.1 Test Data Sets

Two types of data sets were used to evaluate the presented algorithms. The first type represents still images used to evaluate the resolution mosaic EM algorithm (RM-EM). The second type of test data sets represents image sequences of various scenes for traffic monitoring.

The first type includes three data sets: synthetic images, a magnetic resonance image (MRI), and simulated MR images. The first set of these data consists of two groups of synthetic images. They are created with certain specifications chosen to explore the advantages and disadvantages of the tested algorithms.

The synthetic images allow quantitative comparisons between the different algorithms, since the ground truth of the segmentation is known a priori. Each one of the synthetic images of both groups is of size 100×100 pixels and consists of four different classes. Each class is created by four Gaussian distributions with mean values 50, 100, 150, and 200. The layouts of the classes are chosen in a way that different types of edges and corners can appear, which are interpreted as difficulties for the segmentation process. The images in the first group are created so that the classes are set in quadratic-chess form as shown in Fig. 9.2. In the second group the images are generated by two Gaussian distributions superimposed by two other Gaussian distributions as thin and thick lines as shown in Fig. 9.3. All the classes in an image are given the same standard deviation. This can be interpreted as the level of noise added to the image. Obviously, as the noise level increases, the difficulty of the segmentation process increases too. Therefore, three noise levels were used, ranging from low to very high. The standard deviations used are 10, 15 and 20. For each noise level an image in each group is created. The histograms are displayed in the same figures of the synthetic images. This helps to give an estimation of the increasing difficulties to the segmentation process as the noise level increases.

The Gaussian distributions used to create the images of Figs. 9.2 and 9.3 are displayed in Fig. 9.1. It shows that with increasing noise level the overlapping areas between the distributions are also increased and the probability of error is increased too. The overlapped area between two classes is counted as Bayes error. A classifier, such as the minimum distance classifier, cannot classify correctly the pixels lying in this area.



(a) (b) (c)

Figure 9.1: Gaussian distributions in a mixture model used for the synthetic images. Mixture with (a) std = 10. (b) std = 15. (c) std = 20.

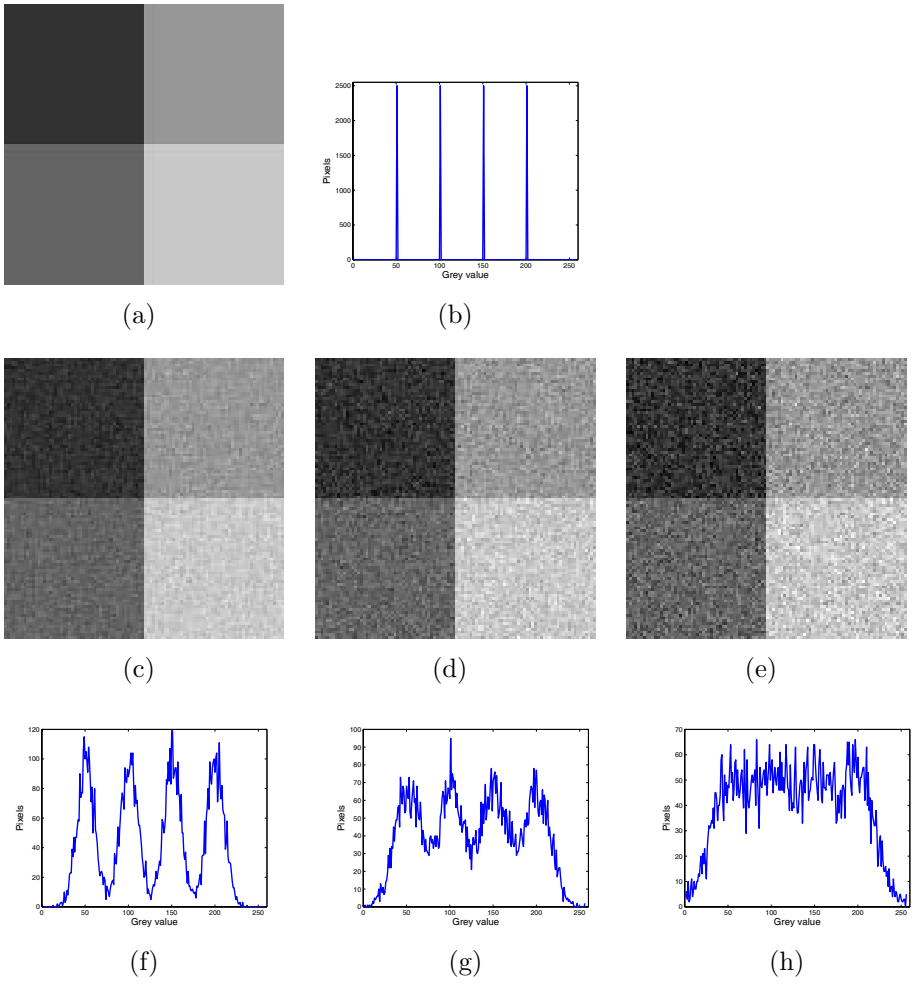


Figure 9.2: Synthetic quadratic images generated by four Gaussian distributions with mean values 50, 100, 150, and 200 and the associated histograms. (a) and (b) Without added noise. (c) and (f) With added noise std = 10. (d) and (g) std = 15. (e) and (h) std = 20.

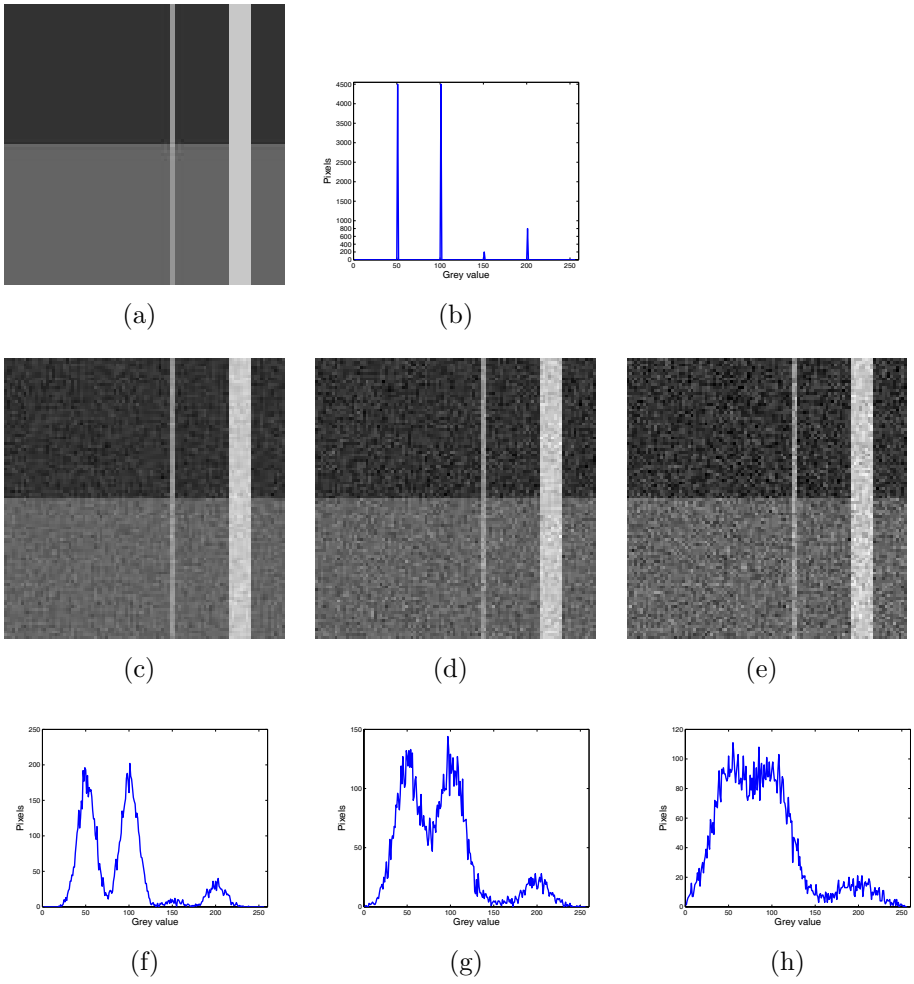


Figure 9.3: Synthetic line images generated by four Gaussian distributions with mean values 50, 100, 150, and 200 and the associated histograms. (a) and (b) Without added noise. (c) and (f) With added noise std = 10. (d) and (g) std = 15. (e) and (h) std = 20.

The second data set is a real magnetic resonance image (MRI) of the human brain. Magnetic resonance images represent the intensity variation of radio waves generated by biological systems when exposed to radio frequency pulses. The image is representing a cross-sectional slice of the target. It can be divided into three regions other than the background. The first region represents the white matter (WM) of the brain tissue, the second the grey matter (GM), and the third region represents the cerebrospinal fluid (CSF) [AU96, UA96]. In MRI many fine features appear, such as edges or boundaries between different regions. Fig. 9.4 shows a real MR image of size 206×167 and the three different tissues. The colour of the CSF is the same colour as the background. Therefore, they are segmented together in the same class.

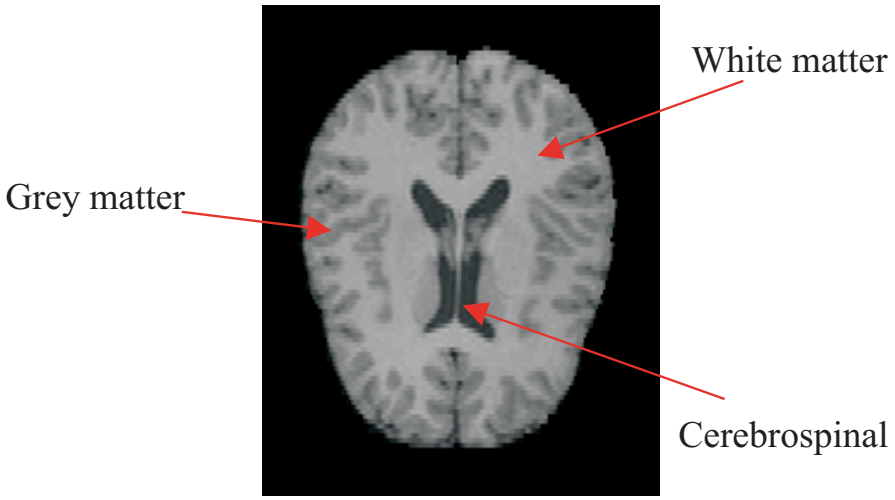


Figure 9.4: Real magnetic resonance image of the human brain.

The third data set consists of simulated MR images. The resulted segmented image by applying the EM algorithm on the real MRI is used as a labelled image to create the images belonging to this data set. Each pixel in a simulated MR image is generated by the Gaussian distribution of the class of the corresponding pixel in the label image. Again three values of standard deviations are used to create three test images, namely, 10, 15, and 20 to represent low, medium, and high noise level, respectively. Fig. 9.5 shows the created images and their associated histograms. This data set is used because it is not possible to produce quantitative segmentation results for the MRI because of the absence of the ground truth. Furthermore, its structure is very difficult to simulate by synthetic images.

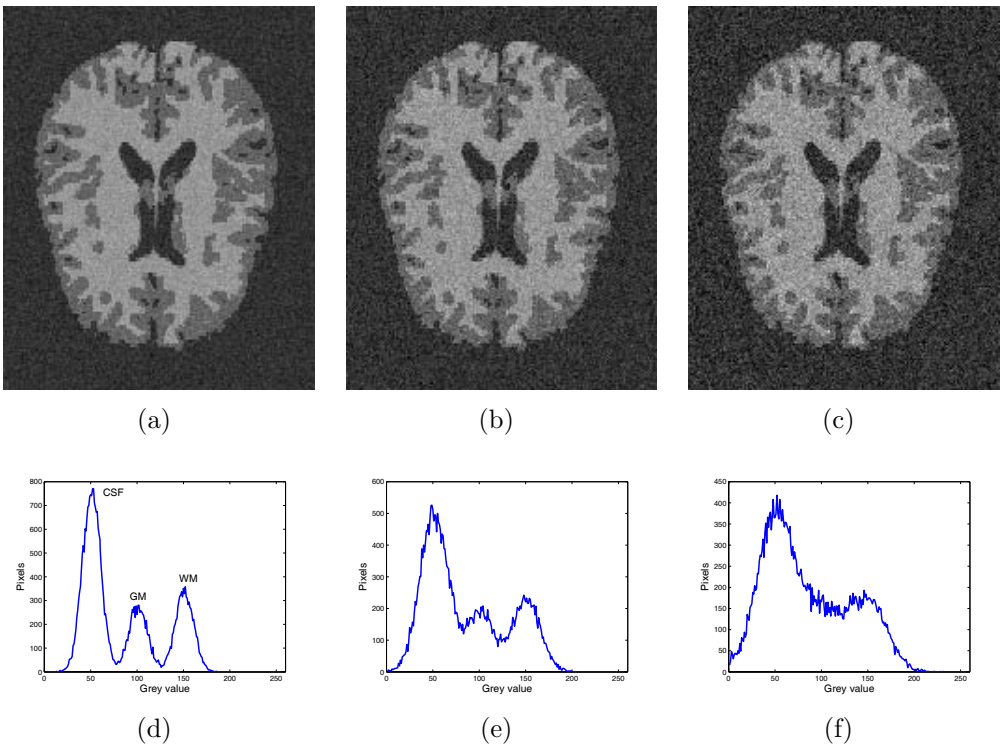


Figure 9.5: Simulated MRI generated by four Gaussian distributions with mean values 50, 100, 150, and 200 and the associated histograms. (a) and (d) std = 10. (b) and (e) std = 15. (c) and (f) std = 20.

The second type of test data is chosen to test and evaluate the algorithms proposed for image sequence segmentation. For this, the application *traffic monitoring* is chosen. In this application it is important to decide which part of the image is an active traffic area, and to detect there moving objects. Eighteen data sets are used consisting of 7 different scenes and containing 1786 frames as well as 9219 moving objects. All the data sets were captured using a stationary video camera. A summarised description of all data sets can be found in Tab. 9.1. They can be categorised into four groups: 1) a front view with a small camera observation angle to the street and only few moving objects, 2) wide view of an intersection with many types of traffic objects, 3) an overview of a busy one way road during bad or varying lighting conditions and 4) an overview acquisition of pedestrians in the campus Vaihingen of Stuttgart University.

Table 9.1: Description of the data sets used for the evaluation of the video segmentation algorithms.

Set No.	Prefix name	Seq. size	Frame/ second	Frame size	Image type	No. of objects	Group
1	Adlershof1	12	25	288×352	Jpeg	13	1
2	Adlershof2	16	25	288×352	Jpeg	11	1
4	Danziger4	40	6	480×340	Bitmap	62	2
6	Danziger6	64	6	480×340	Bitmap	458	2
7	Danziger7	32	3	480×340	Bitmap	229	2
8	Rudower8	24	25	288×352	Bitmap	22	1
9	Rudower9	48	25	288×352	Bitmap	48	4
10	Frankfurt10	64	25	288×360	Jpeg	170	3
11	Frankfurt11	96	25	288×360	Jpeg	192	3
13	RuskaUfer13	128	25	640×480	Jpeg	240	1
14	RuskaUfer14	128	25	640×480	Jpeg	625	1
15	AdlershofAlt15	46	25	288×374	Bitmap	819	3
16	Stuttgart16	64	5	384×512	Jpeg	434	4
20	Stuttgart20	256	5	384×512	Jpeg	1949	4
23	RuskaUfer23	256	25	640×480	Jpeg	1162	1
24	RuskaUfer24	256	25	640×480	Jpeg	926	1
26	RuskaUfer26	128	25	640×480	Jpeg	676	1
27	RuskaUfer27	128	25	640×480	Jpeg	1183	1

(a) Adlershof

(b) RuskaUfer

(c) AdlershofAlt

(d) Frankfurt

(e) Danziger

(f) Stuttgart

Figure 9.6: Selected scenes used for testing and evaluating the image sequence segmentation algorithms. Prefix names as in Tab. 9.1.

In Figs. 9.6(a) and 9.6(b) examples of the front view observation are given. Figs. 9.6(c) and 9.6(d) show two acquisitions in bad lighting conditions. Fig. 9.6(d) shows a dark scene after a fast change in the illumination due to cloud movement. Fig. 9.6(e) shows a wide-angle acquisition of an intersection with different moving objects and Fig. 9.6(f) shows a view of the pedestrian zone in the campus Vaihingen of Stuttgart University.

9.2 Segmentation Evaluation

9.2.1 Evaluation Methods

A common problem in the analysis of segmentation results is the absence of standard evaluation methods and standard test data. Due to the fact that no single segmentation technique is useful for all applications, and different techniques are not equally suited for a particular application, an effective evaluation of the segmentation is very important. It is useful and necessary for selecting the most appropriate technique for a specific application, and furthermore for an optimal parameter setting of the selected technique.

Generally, the segmentation evaluation methods can be classified into *subjective* evaluation methods and *objective* evaluation methods. The subjective methods are based on asking many observers one or more questions after displaying the results of the segmentation. The questions investigate subjectively the segmentation quality, i.e., the evaluation is based on human intuition or judgement. Such evaluation is necessary to study and characterise the perception of different artefacts on the overall quality [GEKS06]. A significant number of observers is required to produce statistically relevant results. This makes subjective evaluation a time-consuming and expensive process [CGE02]. Even this subjective testing presents practical problems since the procedure for comparison and ranking segmentation qualities is not standardised [VMS99]. Thus, for practical reasons, a fair subjective evaluation is not possible in most cases.

Objective evaluation refers to an automatic procedure that assigns the quality of segmentation either in terms of algorithm design *analytically* or in terms of the quality of the results *empirically*. Analytical evaluations have been given in the chapters where the algorithms were introduced. Performance measures are needed for the empirical evaluation. The selection of such measures depends on the application of the segmentation and on the used technique.

9.2.2 Performance Measures for Image Segmentation

The overall accuracy and the accuracy and precision of each class are the performance measures chosen for the first application, the segmentation of still images. These are statistical measures of the error in assigning classes to the processed pixels. The confusion matrix is used to compute these measures. The confusion matrix for a four-class classifier is given in Tab. 9.2.

Table 9.2: Contents of a confusion matrix with C1, C2, C3, and C4 as classes.

		Predicted			
		C1	C2	C3	C4
Actual	C1	(1,1)	(1,2)	(1,3)	(1,4)
	C2	(2,1)	(2,2)	(2,3)	(2,4)
	C3	(3,1)	(3,2)	(3,3)	(3,4)
	C4	(4,1)	(4,2)	(4,3)	(4,4)

The entries in the confusion matrix have the following meaning:

1. The rows x represent the ground truth (actual classification), the columns y represent the predicted classification.
2. The entries in the main diagonal represent the number of correctly classified pixels.
3. The entries of the form (x, y) such that $x \neq y$, represent the number of pixels which were wrongly classified as belonging to class y, while they actually belong to class x. For example the entry in position $(2, 3)$ represents the number of pixels classified as belonging to class 3 while they actually belong to class 2.

The following information can be calculated from the confusion matrix:

1. Accuracy of a class:
$$AC_x = (x, x)/\sum_y (x, y) \tag{9.1}$$
2. Precision of a class:
$$P_x = (x, x)/\sum_x (x, y) \tag{9.2}$$
3. Overall accuracy:
$$AC = \sum_x (x, x)/\sum_x \sum_y (x, y) \tag{9.3}$$
4. Overall error rate:
$$ER = \sum_x \sum_{y, y \neq x} (x, y)/\sum_x \sum_y (x, y) \tag{9.4}$$

A confusion matrix is computed for each image used to test the algorithms EM and the RM-EM (except the real MRI). Then the overall accuracy and the accuracy of each class are computed.

9.2.3 Performance Measures for Video Segmentation

Different performance measures are used to evaluate the results of the algorithms used for the segmentation of image sequences. Some of them are used to evaluate the detection of moving objects and the others are used to evaluate the extraction of active traffic areas.

For the first purpose, the results of the segmentation are blobs, which represent the regions with moving objects determined by the algorithm. The smallest bounding boxes were drawn around the extracted regions. Then they were checked and counted manually by a human operator. Three performance measures were computed:

1. False alarms (FA):
   The number of extracted regions that contains no active moving objects.

2. Missed objects (MO):
   The number of active moving objects that are not included in any extracted regions.

3. Delayed detections (DD):
   The number of active moving objects that are delayed for the first time detection, i.e., the objects that have never been included in any extracted regions yet.

The delayed detections is always a part of the missed objects, but it is computed independently. It can be used to find out how long it takes the algorithm to detect a new moving object that enters the current scene.

The error measures are given in percentage, relative to the number of bounding boxes in the case of false alarms, or relative to the number of moving objects in the case of the other two performance measures (missed objects and delayed detections).

The evaluation of the algorithm for the extraction of the active traffic area is done based on a ground truth. An ideal segmentation for each data set is done manually and used as reference segmentation, as shown, e.g., in Fig. 9.7. The results of the evaluation are the disparity between the reference segmentation (reference mask) and the determined segmentation results (estimated mask).

(a) (b)

Figure 9.7: (a) One input image of an image sequence. (b) Manually segmented active traffic area.

The measure of how close the estimated mask resembles the reference mask is called *spatial accuracy*. It is computed using two additional error measures: the *over segmentation* and the *under segmentation* [JBM+00, GEKS06]:

1. Over segmentation (OS):
   Pixels estimated to belong to the foreground (active traffic area) but belonging to the background in the reference mask.

2. Under segmentation (US):
   Pixels estimated to belong to the background but belonging to the foreground in the reference mask.

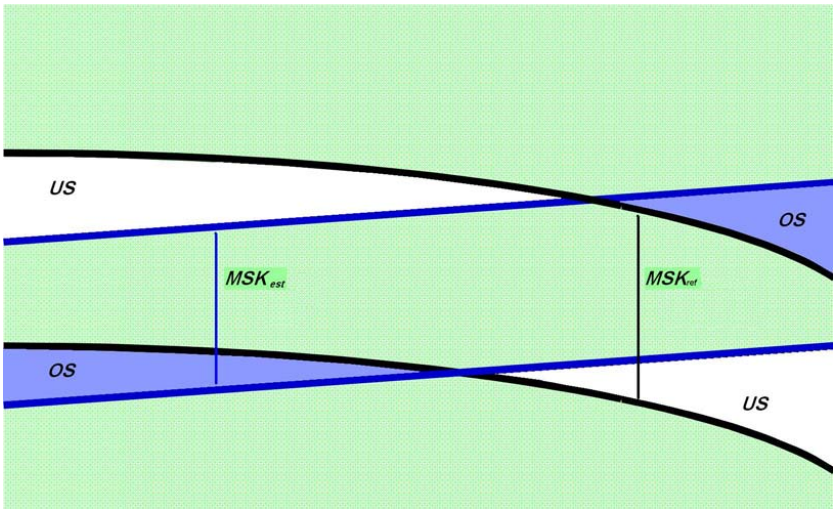


Figure 9.8: Errors produced by over segmentation (OS) and under segmentation (US). MSK_{est}: Estimated mask. MSK_{ref} Actual mask.

Fig. 9.8 illustrates these error measures, where MSK_{est} and MSK_{ref} are the estimated and the reference masks of the active traffic area, respectively.

In some work over and under segmentation are referenced as added background and missing foreground, respectively [VMS99]. Here, we interpret them as false estimated foreground and as false estimated background, respectively. The area of the estimated mask MSK_{est} without the over segmentation OS corresponds to a correct classification of the foreground. The area of the estimated background BG_{est} without the under segmentation US is the correct classification of the background. Tab. 9.3 puts this interpretation in the form of a two-class confusion matrix.

Table 9.3: Contents of the two-class confusion matrix evaluating the extraction of the active traffic area.

		Determined	
		Background	Foreground
Actual	Background	$BG_{\text{est}} - US$	OS
	Foreground	US	$MSK_{\text{est}} - OS$

The spatial accuracy can then be computed either as an absolute value (number of missed pixels) or as a relative value yielding a relative spatial quality measure. In this work the relative spatial accuracy is measured, as the precision of the foreground, i.e., the area of the estimated mask without the over segmentation, relative to the area of the estimated mask (MSK_{est}):

$$\text{Precision} = \frac{MSK_{\text{est}} - OS}{MSK_{\text{est}}} \tag{9.5}$$

For two reasons this gives a more sensitive and realistic measure of the segmentation error than the overall accuracy relative to the size of the mask image. First, the size of the foreground is usually small compared to the size of the mask image. For many data sets the greatest part of the mask image is background. So, dividing by the mask image size may give too optimistic results. Second, the results may depend on the focus and the resolution of the camera. In other words, one may get different results for the same scene if the acquisition conditions change. However, this error measure does not take into consideration the error of under segmentation.

9.3 Results of the Resolution Mosaic Image Segmentation

The RM-EM algorithm was applied on the three test sets of the first data type. The results in terms of segmented images and segmentation accuracies are compared to the results of the conventional EM algorithm.

Fig. 9.9 shows the segmentation results applying the algorithms to the synthetic quadratic images, in the left column the conventional EM algorithm, in the right column the RM-EM. The images are placed side by side to enable visual comparison of an equal noise level.

One can easily observe that the conventional EM algorithm misclassifies many pixels, although, all the surrounded pixels are correctly classified. So, the EM algorithm fails to utilise the strong spatial correlation between neighbouring pixels. This is due to the Gaussian mixture model topology, which assumes that all pixels are independently and identically distributed. However, it has an advantage in that it reduces the computational complexity of the segmentation task by allowing the use of the well-characterised Gaussian density function [Sae97].

For the conventional EM algorithm and for the RM-EM the confusion matrices of the images and the overall accuracy of the segmentation are displayed in Tab. 9.4.

The accuracies given in the last table show that the RM-EM is much less sensitive to noise in a comparison to the conventional EM algorithm. As the noise level increases from 10 to 20 the overall accuracy of the segmentation by the EM algorithm drops from 99% to 82%. For the RM-EM algorithm it decreases only by 2%.

The segmentation results of the RM-EM algorithm using the synthetic line images are displayed in the Figs. 9.10(b), 9.10(d), and 9.10(f). The results of the conventional EM algorithm are displayed side by side in Figs. 9.10(a), 9.10(c), and 9.10(e). Visually, the segmentation of the thin line by the conventional EM is better. This could be expected since the new algorithm is supposed to process fine features like edges in high resolution. But this depends on the edge detection step, which may fail to detect parts of the edges. On the other side, the following statistical results show that the segmentation of the thin line with increasing noise level is more reliable using the RM-EM algorithm.

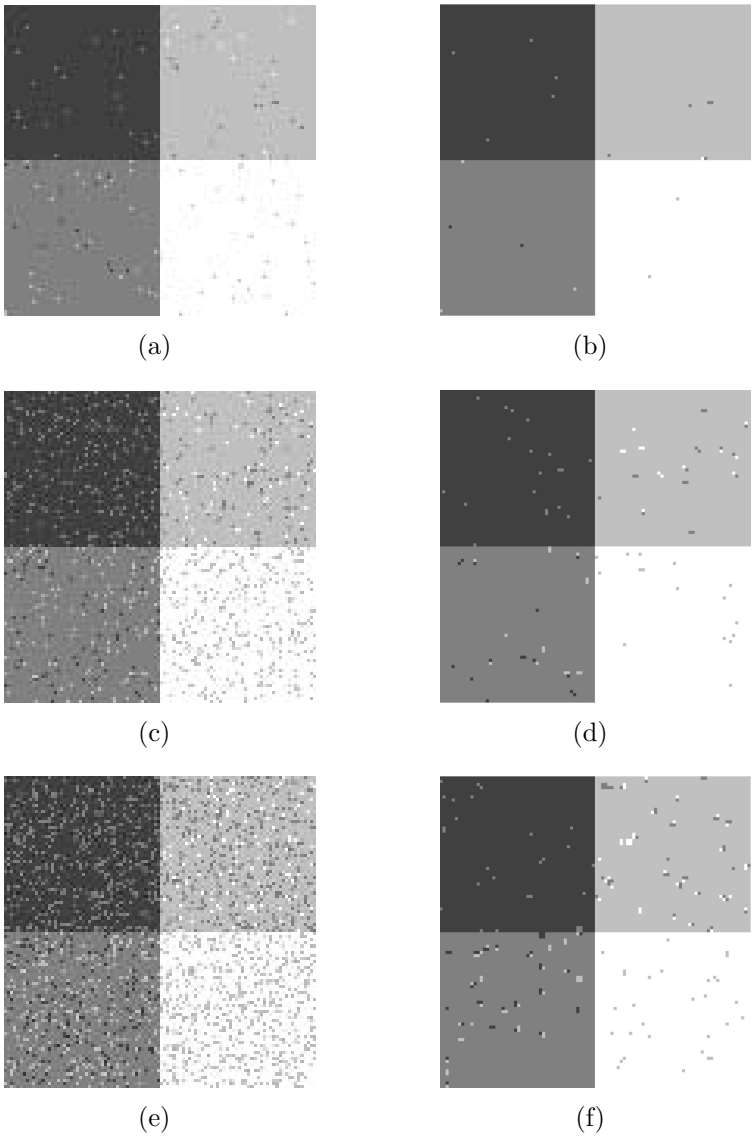


Figure 9.9: Segmentation results of the synthetic quadratic images by the conventional EM (left) and the RM-EM (right) algorithm. Images with (a) and (b) std = 10. (c) and (d) std = 15. (e) and (f) std = 20.

Table 9.4: Confusion matrices of the segmentation results of the synthetic quadratic image with different noise levels.

EM - std = 10

	C1	C2	C3	C4
C1	2480	20	0	0
C2	14	2460	26	0
C3	0	20	2471	9
C4	0	0	25	2475
			AC =	98.86%

RM-EM - std = 10

	C1	C2	C3	C4
C1	2495	5	0	0
C2	2	2495	3	0
C3	0	5	2494	1
C4	0	0	2	2498
			AC =	99.82%

EM - std = 15

	C1	C2	C3	C4
C1	2308	192	0	0
C2	58	2331	111	0
C3	0	100	2344	56
C4	0	0	319	2181
			AC =	91.64%

RM-EM - std = 15

	C1	C2	C3	C4
C1	2481	19	0	0
C2	15	2470	15	0
C3	0	17	2470	13
C4	0	0	16	2484
			AC =	99.05%

EM - std = 20

	C1	C2	C3	C4
C1	2073	427	0	0
C2	169	2096	235	0
C3	0	347	2048	105
C4	0	0	558	1942
			AC =	81.59%

RM-EM - std = 20

	C1	C2	C3	C4
C1	2472	28	0	0
C2	36	2427	37	0
C3	0	44	2419	37
C4	0	0	30	2470
			AC =	97.88%

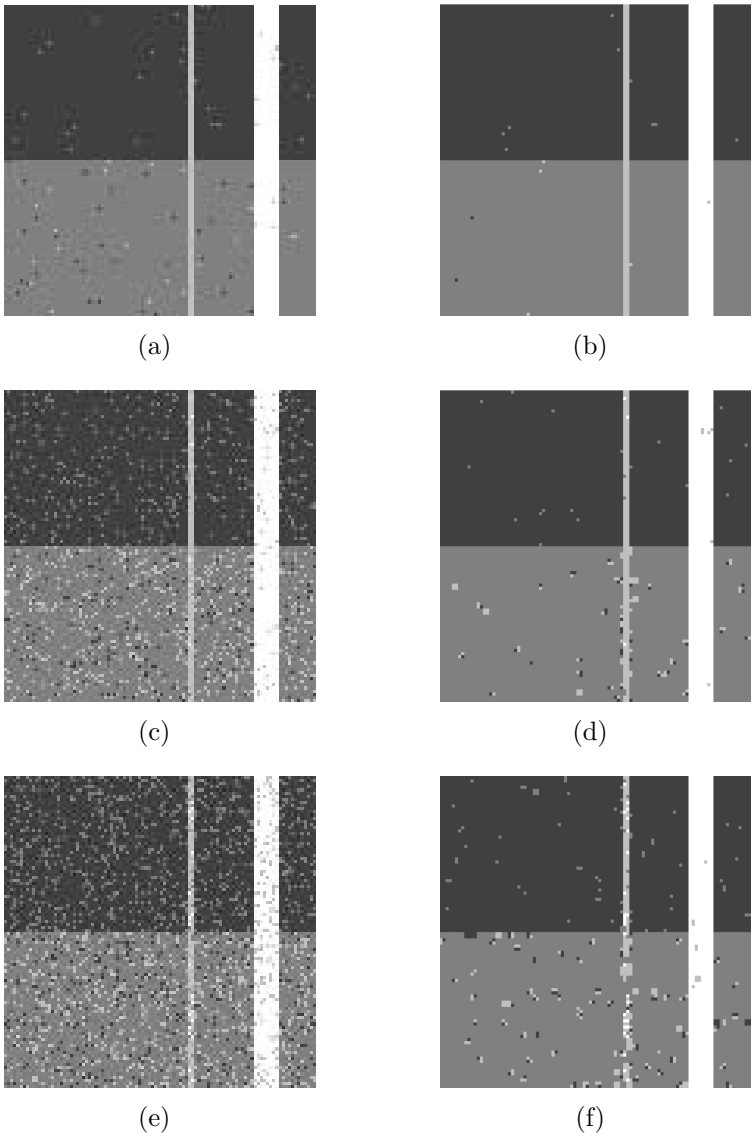


Figure 9.10: Segmentation results of the synthetic line images by the conventional EM (left) and the RM-EM (right) algorithm. Images with (a) and (b) std = 10. (c) and (d) std = 15. (e) and (f) std = 20.

The overall accuracies of the conventional EM and the RM-EM algorithm for the synthetic line images are very close to the results for the synthetic quadratic images. Therefore, only the summarised overall accuracies are given in Tab. 9.5.

Table 9.5: Overall accuracies for the synthetic line images.

	std=10	std=15	std=20
EM	99.12%	89.82%	82.35%
RM-EM	99.84%	98.51%	96.17%

The particular important classes of the synthetic line images are the classes of the thin and thick line. Consequently, a two-class confusion matrix is computed, where the subject class is categorised as positive (+ve) and all other classes are gathered in one category as negative (-ve). The precision of Eq. 9.2, as a performance measure for both classes, is computed and displayed at the end of each table. Tabs. 9.6 and 9.7 show the confusion matrices and the precisions of the thin and thick line classes for all noise levels obtained by the conventional EM and the RM-EM algorithm.

Table 9.6: Confusion matrices of the segmentation results for the thin line class.

EM	std = 10		std = 15		std = 20	
	- ve	+ ve	- ve	+ ve	- ve	+ ve
- ve	9771	29	9285	515	9075	725
+ ve	0	200	8	192	42	158
Precision	87.34%		27.16%		17.89%	

RM-EM	std = 10		std = 15		std = 20	
	- ve	+ ve	- ve	+ ve	- ve	+ ve
- ve	9795	5	9716	84	9648	152
+ ve	0	200	7	193	70	130
Precision	97.56%		69.68%		46.10%	

Table 9.7: Confusion matrices of the segmentation results for the thick line class.

EM	std = 10		std = 15		std = 20	
	- ve	+ ve	- ve	+ ve	- ve	+ ve
- ve	9200	0	9195	5	9180	20
+ ve	9	791	39	761	136	664
Precision	100%		99.35%		97.08%	

RM-EM	std = 10		std = 15		std = 20	
	- ve	+ ve	- ve	+ ve	- ve	+ ve
- ve	9200	0	9197	3	9172	28
+ ve	1	799	6	794	10	790
Precision	100%		99.62%		96.58%	

The results of the new segmentation algorithm for the thin line class are much better than for the conventional EM algorithm, while they are at least equally good for the thick line class. This shows that the resolution mosaic improves the results in case of fine edges and high noise levels.

Transformation of the images into the resolution mosaic images can be considered as a pre-processing step. It can be used to simplify the computation of the EM algorithm by reducing the size of the input image. For example, the size of the high noisy synthetic quadratic image is reduced to less than 15% of the original size. It can be used as well to enhance the accuracy of the estimations of the unknown parameters of the Gaussian mixture model by the EM algorithm by performing local noise suppression. Tab. 9.8 displays the estimated mean values of the distributions of the mixture using the conventional EM algorithm and with the pre-processing step of the resolution mosaic (RM-EM), where the actual vector of the mean values is [50 100 150 200]. The errors displayed in the table are computed as the average of the relative error of the estimated mean value of each class. It is clear from the table that the step of the resolution mosaic helps to have a stable accuracy of the estimation of the unknown parameters of the mixture against the increase of the noise level.

Tab. 9.9 displays the estimated standard deviations of the Gaussian mixture by the conventional EM algorithm and with the pre-processing step of the resolution mosaic (RM-EM). Almost all the estimations (with only two exceptions) after the resolution mosaic step are less than the actual values of the standard deviations. This can be explained as follows: this step reduces the deviations between the neighbouring pixels and hence within the concerned class. In other words, it depresses the noise locally within a neighbourhood. However, as the size and the uniformity of the class increase, the possibility of a better "denoising" increases. This is the case for all classes of the quadratic images and the background as well as for the thick line class of the line images. For the thin line class the results are reversed.

Table 9.8: Estimated mean values of the Gaussian mixture model using the EM and the RM-EM algorithm.

		Quadratic images		line images	
	σ_{act}	$[\mu_{\text{est},1}\ \mu_{\text{est},2}\ \mu_{\text{est},3}\ \mu_{\text{est},4}]$	Err	$[\mu_{\text{est},1}\ \mu_{\text{est},2}\ \mu_{\text{est},3}\ \mu_{\text{est},4}]$	Err
EM	10	[50.1 99.6 150.1 200.3]	0.2%	[50.1 100.2 150.8 200.4]	0.3%
	15	[48.1 98.0 155.3 203.9]	2.8%	[48.9 98.3 137.4 202.3]	3.4%
	20	[46.1 96.6 158.1 207.1]	5.0%	[46.6 96.1 151.2 205.1]	3.5%
RM-	10	[49.9 99.8 150.3 200.7]	0.2%	[50.2 99.5 150.1 201.4]	0.4%
EM	15	[48.8 100.6 149.9 200.4]	0.8%	[50.4 99.3 144.8 203.1]	1.6%
	20	[49.9 100.9 149.7 201.4]	0.5%	[49.9 101.1 143.9 198.3]	1.6%

The displayed symbol $\mu_{\text{est},i}$ stands for the estimated mean of class i and σ_{act} stands for the actual standard deviation.

The second data set used to test the proposed algorithm is a real MRI. Fig. 9.11 illustrates the segmentation using the conventional EM and the RM-EM algorithm. It is quite difficult to find visual differences. With the naked eye one cannot evaluate the both results. Computationally, there are differences between both algorithms due to the reduction of the input size and hence the reduction of the number of iterations of the EM algorithm. The input MRI has the dimension 206×167 of grey level pixels, which is $34,402$ pixels. Using the resolution mosaic, the size of the input image is reduced to $8,917$ pixels, which is about 26% of the size of the original image. The number of iterations needed by the EM algorithm to reach a conversion of the estimation of the mixture parameters is 737, while it is 25 for the list created by the resolution mosaic algorithm.

Table 9.9: Estimated standard deviations values of the Guassian mixture model using the EM and the RM-EM algorithm.

		Quadratic images		line images	
	σ_{act}	$[\sigma_{\text{est},1}\ \sigma_{\text{est},2}\ \sigma_{\text{est},3}\ \sigma_{\text{est},4}]$	Err	$[\sigma_{\text{est},1}\ \sigma_{\text{est},2}\ \sigma_{\text{est},3}\ \sigma_{\text{est},4}]$	Err
EM	10	[9.9 10.4 10.4 9.5]	3.5%	[9.9 9.9 10.7 10.3]	3.0%
	15	[13.9 17.9 20.7 13.2]	19%	[14.5 15.8 27.7 14.4]	24%
	20	[17.7 22.8 23.3 16.7]	14%	[18.1 21.9 34.5 17.9]	25%
RM-	10	[7.0 8.8 7.0 5.9]	28%	[7.6 7.4 10.4 8.9]	16%
EM	15	[10.5 10.8 8.9 11.9]	30%	[13.6 7.0 16.7 8.1]	29%
	20	[15.3 10.7 10.7 15.0]	35%	[16.3 12.2 17.5 14.6]	24%

The displayed symbol $\sigma_{\text{est},i}$ stands for the estimated standard deviation of class i and σ_{act} stands for the actual standard deviation.

Similar to the synthetic images, the segmentation results of the simulated MR images are given as figures in Fig. 9.12 and confusion matrices in Tab. 9.10.

Using the EM algorithm one can note that for all noise levels there are many misclassified pixels in the background, which again reminds us that it fails to utilise the strong spatial correlation between pixels. Visually, the RM-EM algorithm gives bad segmentation for edge pixels, e.g., for the edges between the class of the grey matter and the background or between the grey white and the white matter. The accuracy of the segmentation depends on the creation of the mask images and hence the creation of the mosaic map as shown in Fig. 5.4 and in Section 5.2.2. As much as the mask images are precise, the segmentation of the edge pixels and the thin classes is accurate.

The statistical results show in contrast that the RM-EM algorithm gives better results for the class of the grey matter than the conventional EM algorithm under all noise levels. They show also much more stable overall accuracy if the noise level is increased.

The grey matter class (class 2) as shown in Fig. 9.4 has a special structure with many edges and overlaps with the background as well as with the white matter. Therefore, in Tab. 9.11 the segmentation precisions of this class for all noise levels using the conventional EM algorithm and the RM-EM algorithm are given.

Table 9.10: Confusion matrices of the segmentation results of the simulated MRI with different noise levels.

EM - std = 10

	C1	C2	C3
C1	18800	193	0
C2	41	6892	18
C3	0	83	8375
		AC =	99.03%

RM-EM - std = 10

	C1	C2	C3
C1	18921	72	0
C2	70	6805	76
C3	0	66	8392
		AC =	99.17%

EM - std = 15

	C1	C2	C3
C1	17677	1316	0
C2	245	6614	92
C3	0	1078	7380
		AC =	92.06%

RM-EM - std = 15

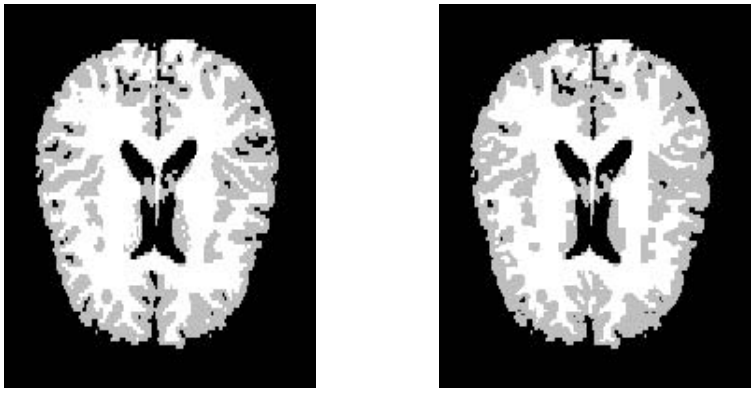
	C1	C2	C3
C1	18469	524	0
C2	186	6628	137
C3	0	413	8045
		AC =	96.34%

EM - std = 20

	C1	C2	C3
C1	17856	1137	0
C2	1246	5552	153
C3	2	2636	5820
		AC =	84.96%

RM-EM - std = 20

	C1	C2	C3
C1	18511	476	6
C2	412	6046	493
C3	6	483	7969
		AC =	94.55%



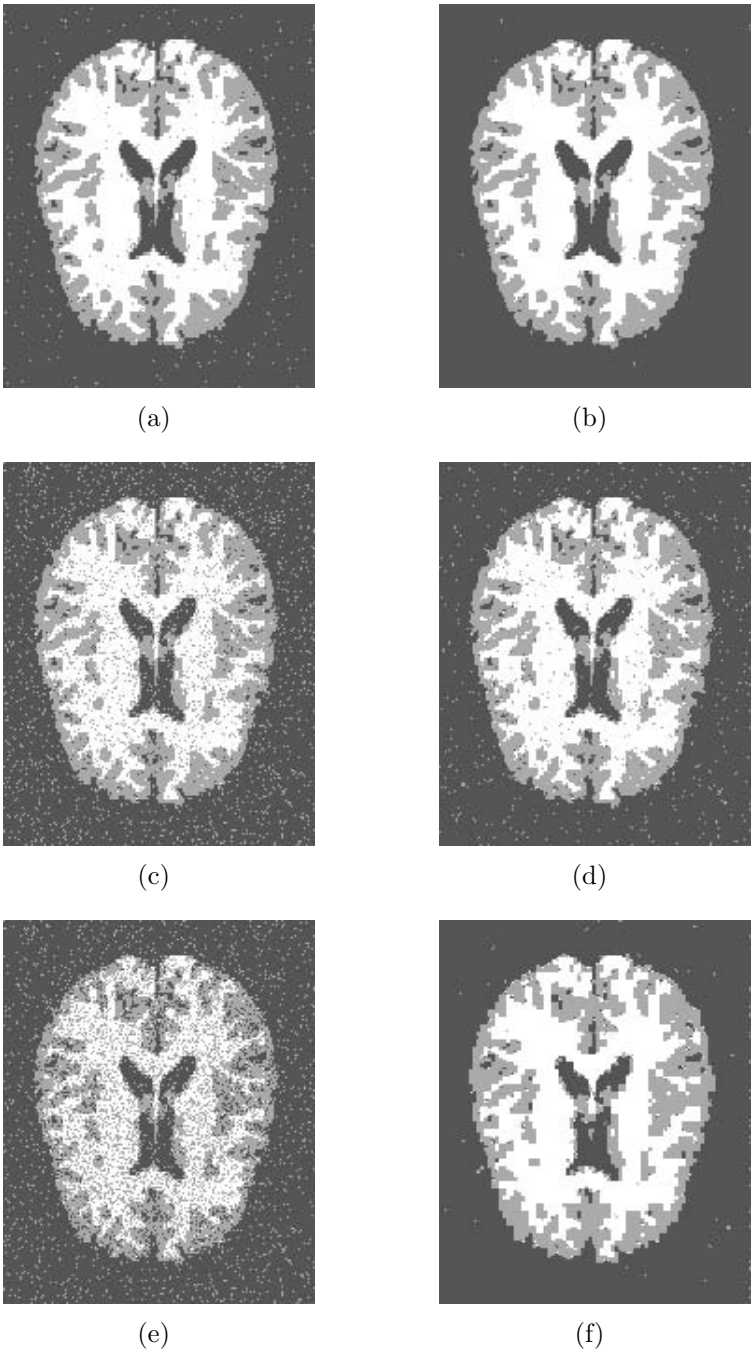
(a) (b)

Figure 9.11: Segmentation results of the real MRI by (a) conventional EM. (b) RM-EM.

The results show stable performance of the RM-EM algorithm over the conventional EM algorithm. The precision of the EM algorithm decreases from 96% to 60% as the noise level increases from std = 10 to std = 20. The decrease in the results of the RM-EM algorithm is from 98% to 86% for the same increase of the noise level.

Table 9.11: Precision for the grey matter class of the simulated MR images.

	std=10	std=15	std=20
EM	96.15%	73.42%	59.54%
RM-EM	98.01%	87.61%	86.31%



(a) (b)

(c) (d)

(e) (f)

Figure 9.12: Segmentation results of the simulated MR images by the conventional EM (left) and the RM-EM (right) algorithm. Images with (a) and (b) std = 10. (c) and (d) std = 15. (e) and (f) std = 20.

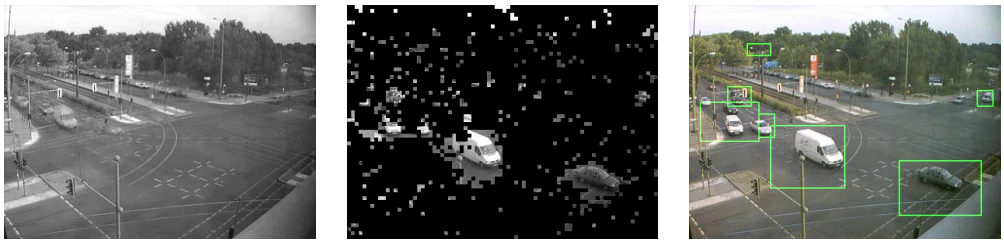
9.4 Results of the Wavelet-based Video Segmentation

9.4.1 2D Wavelet-based Video Segmentation

It is one subject of this thesis to compare the proposed algorithms for different application areas. To do this for the 3D wavelet-based segmentation algorithm the 2D wavelet-based algorithm [TCAA05] has to be considered first.

The updating parameter α and the scaling factor β of the Eqs. 2.25 and 2.27 need to be given by an operator before the 2D wavelet-based algorithm can be applied. The data sets of the *Adlershof* scene have been chosen to set these parameters because of their stable lighting conditions, homogeneous background, and simple scenes.

To estimate the best values for α and β a black spot of size 20×20 pixels was added to the first frame outside the active traffic area. The time required to find a good background estimation was measured. The background is initialised by the first frame of the sequence. Generally, the best estimations were found for values of $0.9 \leq \alpha \leq 0.95$ and value of $\beta = 5$.



(a) (b) (c)

Figure 9.13: Results of the 2D wavelet-based algorithm for the scene *Danziger*. (a) Estimated background. (b) Extracted ROI. (c) Corresponding bounding boxes.

Next, the algorithm was applied to all data sets. Fig. 9.13 gives an example of the algorithm outputs for the data set *Danziger6* (Fig. 9.6(e)). The estimated background is shown in Fig. 9.13(a). The data set shows crossing objects after stopping for a while in the front of traffic lights. Therefore, slight shadows of moving objects (that are currently at the right end) can be seen at the left end of the estimated background. This is because these objects are considered as a part of the background while they were stopping.

Fig. 9.13(b) shows the extracted ROI as blobs in a black background. One can easily notice the existence of noise all over the scene. This is due to the continuous change in the illumination. Fig. 9.13(c) shows the moving objects in their bounding boxes (BB). Generally, the blobs that do not meet certain criteria are assumed to be noise and are not considered in the computation of the bounding boxes. The criteria can be the area or the ratio of width to height of the blobs. The bounding boxes were counted frame by frame by a human operator. Tab. 9.12 shows the results of the segmentation of moving objects. They are given in terms of false alarm rates (FA) relative to the number of the bounding boxes, and missed object rates (MO) and delayed detection rates (DD) relative to the number of the moving objects.

Table 9.12: Results of the 2D wavelet-based algorithm in terms of extracted bounding boxes.

	BB	FA		MO		DD	
Adlershof1	37	21	56.8%	2	15.4%	2	15.4%
Adlershof2	26	16	61.5%	1	9.1%	1	9.1%
Danziger4	351	222	63.2%	14	22.6%	0	0.0%
Danziger6	377	105	27.9%	115	25.1%	25	5.5%
Danziger7	260	117	45.0%	25	10.9%	4	1.7%
Rudower8	73	56	76.7%	0	0.0%	0	0.0%
Rudower9	40	1	2.5%	9	18.8%	1	2.1%
Frankfut10	101	5	5.0%	54	31.8%	4	2.4%
Frankfut11	147	40	27.2%	70	36.5%	44	22.9%
RuskaUfer13	145	17	11.7%	112	46.7%	44	18.3%
RuskaUfer14	276	16	5.8%	298	47.7%	78	12.5%
AdlershofAlt15	241	14	5.8%	387	47.3%	234	28.6%
Stuttgart16	171	71	41.5%	293	67.5%	114	26.3%
Stuttgart20	3196	2654	83.0%	1168	59.9%	295	15.1%
RuskaUfer23	506	90	17.8%	491	42.3%	81	7.0%
RuskaUfer24	334	67	20.1%	401	43.3%	101	10.9%
RuskaUfer26	240	33	13.8%	329	48.7%	3	0.4%
RuskaUfer27	428	51	11.9%	662	56.0%	207	17.5%

(a) (b)

Figure 9.14: Results of the 2D wavelet-based algorithm for the scene *Frankfurt*. (a) and (b) Bounding boxes in two successive frames in changing lighting conditions.

(a) (b)

Figure 9.15: Results of the 2D wavelet-based algorithm for the scene *Ruska-Ufer*. (a) Integration of some moving objects in the far view in the estimated background. (b) Bounding boxes show the missed objects in the far view.

For the first and second data sets the algorithm has shown a good performance, especially in the case of missed objects. However, it gives a high rate of false alarms even in stable lighting conditions. The reason may be the small number of frames in these data sets, where the algorithm needs long time for a good background estimation.

Generally, for the data sets of the scene *Danziger* the movement of the trees was the main reason for the high number of false alarms. A comparison between *Danziger6* and *Danziger7* shows that the algorithm tends to miss more objects in a scene with slow motion, since the data set *Danziger7* has a faster sampling rate. Moreover, in many cases the algorithm was not able to detect the new objects as soon as they appear. This explains the high numbers of delayed detections in the data set *Danziger6*.

However, for the data set *Frankfurt11* the error measures false alarms and missed objects increase dramatically due to a fast change in the lighting conditions. The algorithm was not able to adapt its estimation of the background as the scene turns to dark. It has detected all the parts of the scene as moving objects as can be seen in Fig. 9.14. When the scene was very dark the algorithm failed to detect the moving objects and produced a high rate of missed objects.

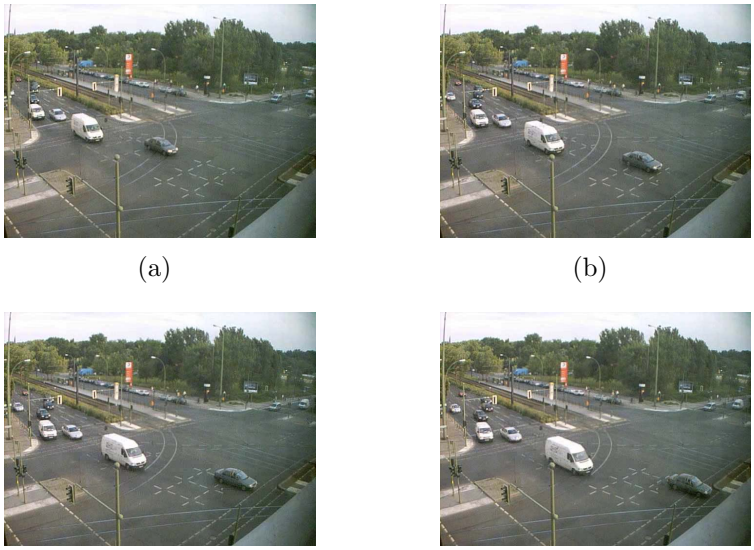
The results of the data sets *RuskaUfer* show reasonable false alarm rates but very high missed object rates. These results can be explained by two reasons, respectively. First, there are not many disturbances in the background and stable lighting conditions. Second, in far views the moving objects move slowly, and therefore they are integrated into the estimated background as shown in Fig. 9.15, although the updating rate was set to 0.95 which allows only very slow integration of a changing situation in the estimated background.

The algorithm gives very bad results for the data set *Stuttgart*. The problem with this data set is that the movement areas of the pedestrians are very small relative to the dimensions of the image. Thus, the grey level changing between the successive frames is in many cases not much greater than the change in illumination of the scene, i.e., the algorithm was not able to detect the movements as ROI. The results show the highest false alarm rates among all the data sets and in the same time the highest missed object rates. Generally, the algorithm is not suitable for such an acquisition.

9.4.2 3D Wavelet-based Segmentation

Extraction of the Regions of Interest

The parameters that had to be set for the proposed algorithm were the size of the masks of the median filter and the size and structure of the dilation operator. By trial and error it was found that good results can be obtained with a 5×5 for the median filter and a 3×3 *diamond* structure for the dilation operator. Three levels of the wavelet analysis were applied to all data sets (an exception was the *Adlershof1* data set with only two levels because of its size). The results of each level are independent of the results of the other two levels.



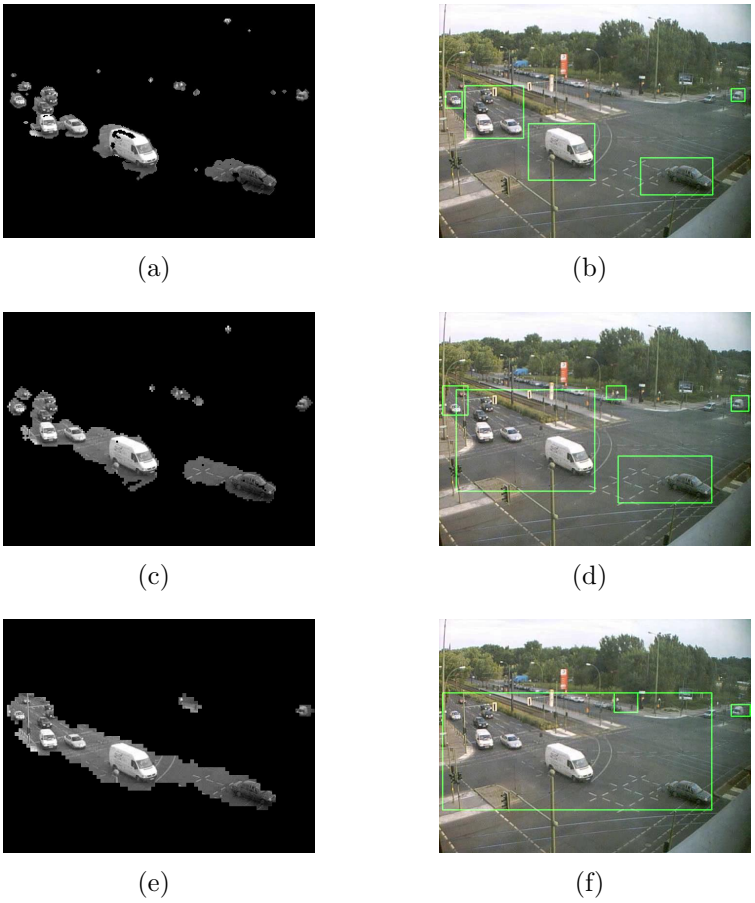
(a) (b)

(c) (d)

Figure 9.16: Selected successive frames that form input groups for the 3D wavelet-based algorithm. Frame number (a) One. (b) Five. (c) Seven. (d) Eight.

The algorithm processes the input sequence in groups of frames. The number of frames in a group depends on the chosen level of analysis. For the first level it is two, for the second and third levels it is four and eight frames, respectively. In general, if j is the chosen level of analysis then 2^j frames are processed together to compute one mask representing the detection of the moving objects, or simply the regions of interest (ROI) in the corresponding group of frames.

Fig. 9.16 shows the first and last frames of each group. The frame in Fig. 9.16(d) will be used often to demonstrate the various results of the algorithm. That is why it has been chosen to represent the last of the group of frames of all analysis levels. The other three frames are selected to show the start frame of the groups at each analysis level. This figure is needed to explain the following results.



(a) (b) (c) (d) (e) (f)

Figure 9.17: Results of the 3D wavelet-based segmentation algorithm for the scene *Danziger*. (a) Extracted ROI. (b) Corresponding bounding boxes for the first analysis level. (c) and (d) Results of the second level. (e) and (f) Results of the third level.

Fig. 9.17 shows an example of the outputs of the algorithm when applied to the data set *Danziger6* for three different analysis levels. Each blob of a ROI in Fig. 9.17(a) represents the detection of motion using the first analysis level in two successive frames, namely the movements in the frames in Figs. 9.16(c) and 9.16(d).

For the second analysis level each blob represents the motion in four successive frames, namely the movements between the frames in Figs. 9.16(b) and 9.16(d). For the third analysis level each blob represents the motion between the frames in Figs. 9.16(a) and 9.16(d).

Tab. 9.13 contains the results obtained using the 3D wavelet-based algorithm. The data sets belonging to the same scene are grouped together.

Table 9.13: Results of the 3D wavelet-based algorithm in terms of extracted bounding boxes.

	Level	BB	FA		MO		DD	
Adlershof	Level 1	26	3	11.5%	0	0.0%	0	0.0%
	Level 2	36	10	27.8%	0	0.0%	0	0.0%
	Level 3	16	5	31.3%	0	0.0%	0	0.0%
Danziger	Level 1	488	11	2.3%	85	11.3%	55	7.3%
	Level 2	606	71	11.7%	17	2.3%	3	0.4%
	Level 3	516	81	15.7%	6	0.8%	6	0.8%
Rudower9	Level 1	56	8	14.3%	0	0.0%	0	0.0%
	Level 2	52	4	7.7%	0	0.0%	0	0.0%
	Level 3	48	0	0.0%	0	0.0%	0	0.0%
Frankfurt	Level 1	315	16	5.1%	58	16.0%	5	1.4%
	Level 2	388	87	22.4%	47	13.0%	2	0.6%
	Level 3	366	115	31.4%	47	13.0%	10	2.8%
AdlershofAlt15	Level 1	288	5	1.7%	122	14.9%	55	6.7%
	Level 2	196	11	5.6%	51	6.2%	12	1.5%
	Level 3	128	6	4.7%	188	23.0%	178	21.7%
RuskaUfer	Level 1	2888	714	24.7%	376	7.8%	116	2.4%
	Level 2	2381	582	24.4%	168	3.5%	50	1.0%
	Level 3	1784	468	26.2%	249	5.2%	45	0.9%
Stuttgart	Level 1	1918	388	20.2%	441	17.1%	117	4.5%
	Level 2	1892	348	18.4%	234	9.0%	84	3.2%
	Level 3	1528	284	18.6%	205	7.9%	100	3.9%

For the data sets *Adlershof* the results show perfect detection of the moving objects with zero missed detection and reasonable false alarm rates in the case of the first analysis level. This can be referred to the ideal lighting conditions.

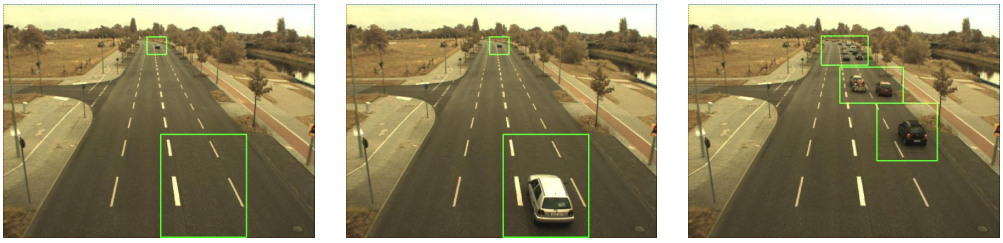
Considering the results of the data sets *Danziger* in Tab. 9.13 and Fig. 9.17 one can conclude that the algorithm avoids many problems of the 2D wavelet-based algorithm. It avoids the problems of the movement of the trees in the background, the disturbances of the reflection of light on the moving objects, and the moving pedestrians out of focus. The results are much better in terms of all statistical error measures. They are improving with the analysis level. That is, the decrease in the rate of the false alarms between the second and third analysis level proofs that the algorithm has lower sensitivity to the small movements and to noise as the analysis level increases. On the other hand, the extracted regions of interest become bigger and bigger than the moving objects which means poorer localisation.

Generally, the second and third analysis levels give no convenient results for the scenes with fast or overlapping moving objects. Post-processing may be needed for object localisation or feature extraction of moving objects.

(a) (b)

Figure 9.18: Results of the 3D wavelet-based algorithm for the scene *Frankfurt*. (a) and (b) Bounding boxes in two successive frames in changing lighting conditions.

The results of the scene *Frankfurt* of the first and second analysis levels show better false alarm rates than those of the 2D wavelet-based algorithm. The results of the third level are a bit worse. However, for all levels the rate of missed objects and the rate of delayed detections are much better compared to the 2D wavelet-based algorithm. We noticed that the 3D wavelet-based algorithm has responded better to the problem of the fast changing in lighting conditions. Fig. 9.18 shows the two frames corresponding to the frames in Fig. 9.14. The bounding boxes show that the algorithm was able to adapt itself while the scene was turning to dark.



(a) (b) (c)

Figure 9.19: Results of the 3D wavelet-based algorithm for the scene *Ruska-Ufer*. (a) False alarm due to an empty bounding box. (b) A new moving object in the successive frame that enters the scene and explains the existence of this bounding box. (c) Detection of moving objects in the far view.

The results of the scene *RuskaUfer* gave higher false alarm rates comparing to the 2D wavelet-based algorithm. For the results of the first level the main reason is the reflection of light on the water canal appearing in the right end of the scene. This type of errors did not appear in the results of the 2D wavelet-based algorithm because of the application of the local thresholding as discussed following Eq. 2.27. This error type does not appear either in the results of the 3D wavelet-based algorithm using the second and third analysis levels. The reason is that in the high analysis levels the images are processed in lower spatial and temporal resolutions and so the fast very local changes (noise) are suppressed. The high false alarm rates in the results of the second and third levels are due to different reasons. The extracted regions represent sometimes the movement that will take place in the future, i.e., they show the movement before the objects appear in the scene, since they represent the movement in a group of frames. This prediction of a movement is counted as false alarm until the moving object appears as illustrated in Fig. 9.19.

The results regarding the rate of missed objects and the rate of delayed detections are much better than those obtained by the 2D wavelet-based algorithm. Fig. 9.19(c) can be compared to Fig. 9.15. The objects in the far view, which have been integrated in the estimated background using the 2D algorithm, are detected even in slow motion.

(a) (b)

Figure 9.20: Results of the 3D wavelet-based algorithm for the scene *Stuttgart*. (a) Many false alarms appear due to the reflections of light. (b) Some stopping pedestrians (circled) are missed by the algorithm.

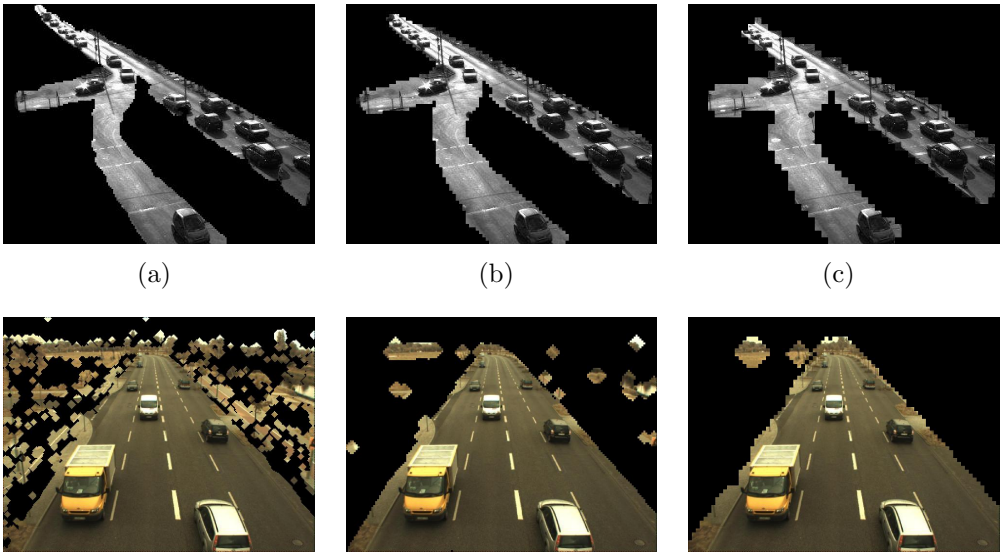
The results of the scene *Stuttgart* are reasonable compared to the 2D wavelet-based algorithm. Using the first level of analysis the false alarm rates and missed object rates are high. The reflection of light is the main reason for the high false alarms. Some pedestrians move very slowly or interrupt their movements and stop for a very short moment so they are accounted as missed objects. Fig. 9.20 shows both cases. We notice here the similarity of the sizes of the bounding boxes around the moving objects and the bounding boxes of the false alarms. Thus, the moving objects have a similar size as the noise, which makes the noise elimination more difficult.

The results of the higher analysis levels are better, especially the missed object rates and the delayed detection rates. The second analysis level gives the best results. It offers a very good compromise between the false alarm rates and the missed object rates.

Extraction of the Active Traffic Area

One of the aims of the proposed algorithm is to give an automatic extraction of the active traffic area. This aim can be achieved simply by cumulating the masks of the ROI of the whole sequence of the input images or of a relative long period. Similar to the extraction of the ROI, three different analysis levels are tested. Fig. 9.21 gives two examples of the results of each level for the two data sets *AdlershofAlt15* and *RuskaUfer24*.

For each data set an ideal segmentation of the active traffic area is done manually and used as reference segmentation. Statistical measures can be obtained for the different levels by comparing the results with the reference segmentation.



(a) (b) (c)

(d) (e) (f)

Figure 9.21: Extracted active traffic area for the scenes *AdlershofAlt* (first row) and *RuskaUfer* (second row) using the (a) and (d) first, (b) and (e) second, (c) and (f) third analysis level.

Tab. 9.14 displays the statistical results for all tested data sets. For each level of analysis the over segmentation (OS) and the under segmentation (US) are given in number of pixels and the precision (Pr) in percentage.

The discussion of these results can be summarised in two observations. The best precisions for the date sets *Adlershof*, *Frankfurt*, and *AdlershofAlt* are obtained for the first level of analysis. The results become worse as the level of the analysis increases. In contrast, the best precisions for the data sets *Danziger* and *RuskaUfer* are obtained for the last level of the analysis. The results become better as the level of the analysis increases. These results can be explained as follows. Basically, the first analysis level is enough to get a good estimation of the active traffic area. As the level of the analysis increases, the extracted ROI becomes larger. Hence, the area added to the estimation increases. Most of these cases result in an over segmentation and increase the error. The results of the first level suffer from noise, which is suppressed in the higher levels. In some cases, the noise suppression is much greater than the over segmentation. Only in the cases *Danziger* and *Ruska-Ufer* the precision of the segmentation increases with the analysis level.

Table 9.14: Results of the 3D wavelet-based algorithm for the extraction of active traffic area in terms of over segmentation (OS), under segmentation (US), and precision (Pr).

	Level 1			Level 2			Level 3		
	OS	US	Pr.	OS	US	Pr.	OS	US	Pr.
Adlershof1	1276	2852	91.5%	2681	1973	84.4%	-	-	-
Adlershof2	1810	383	85.0%	2572	433	79.8%	3803	784	72.1%
Danziger6	12981	7285	82.9%	13696	5668	82.5%	12313	4765	84.1%
Danziger7	13887	7499	81.8%	12788	5896	83.4%	8930	6630	87.7%
Frankfurt10	38080	65030	93.4%	50280	42310	91.7%	70030	65260	88.5%
Frankfurt11	7278	2390	85.3%	10772	1572	80.0%	14661	1653	74.6%
RuskaUfer13	17199	6347	88.6%	9234	4670	93.6%	9220	7632	93.5%
RuskaUfer14	12496	1841	93.9%	11424	5493	94.3%	13135	7908	93.4%
AdlershofAlt15	14710	46550	95.2%	32090	28050	90.6%	48990	30870	86.2%
RuskaUfer23	29626	45725	82.0%	16966	46742	88.7%	18450	44867	88.0%
RuskaUfer24	53082	49917	71.1%	20426	55995	85.9%	13258	54106	90.5%
RuskaUfer26	46461	88817	66.4%	33833	91225	72.5%	28819	85763	76.7%
RuskaUfer27	13880	5588	92.0%	12736	4032	93.3%	12265	3609	93.5%

Figs. 9.21(a), 9.21(b), and 9.21(c) show the decreasing precision with increasing analysis level because of the over segmentation. Figs. 9.21(d), 9.21(e), and 9.21(f) show the opposite case, in which the increasing analysis level enhances the precision because the noise is suppressed.

9.4.3 Different Mother Wavelets

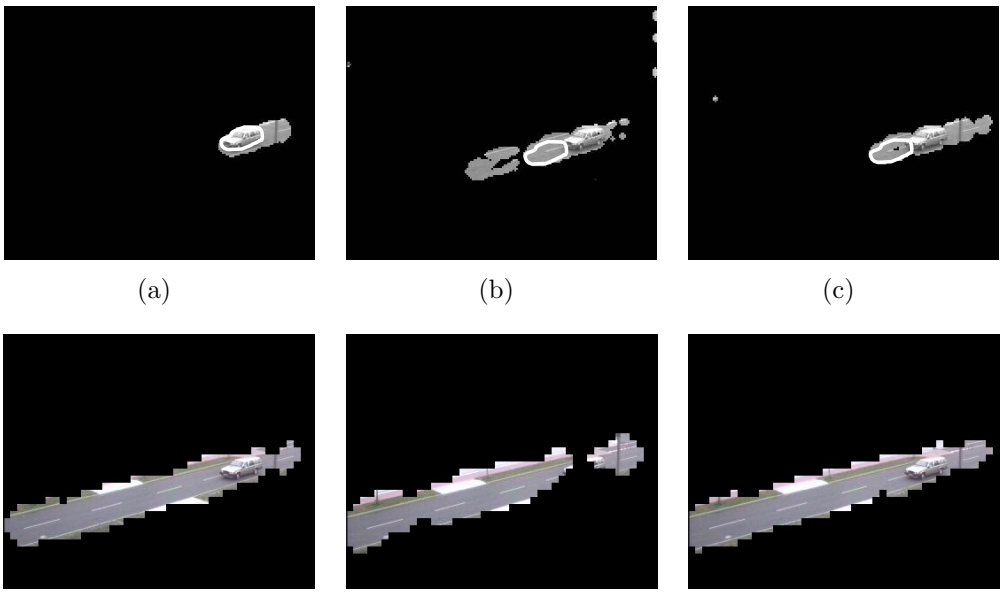
Three mother wavelets from the Daubechies family were tested and evaluated namely, Haar, DB4 and DB8 wavelets. Because all previous results are obtained by the Haar wavelet, its results will be commented here only for the comparison with the other two wavelets.

The data sets *Adlershof* and *Danziger* were used to test and compare the performance of these wavelets, since they represent simple and complex traffic scenes, respectively. As displayed in Tab. 9.15 the number of the extracted bounding boxes as well as the number of false alarms of the mother wavelets DB4 and DB8 are increased dramatically. For the data set *Adlershof* there were delayed detected objects using DB4, third level.

As discussed in Section 6.3 the wavelet analysis is done in an overlapping manner. This leads to the detection of events earlier as they take place. As shown in Figs. 9.22(b) and 9.22(c) a great part of the extracted ROI appears to the left of the moving object. This was not expected since no movement took place yet in these areas. "Too" many pixels from the right areas to the pixels that are processed currently are taken into consideration by the analysis. This leads to a misestimation of these parts. As a result the extracted regions become bigger than the moving objects and are shifted to the left. This is clearly seen by the results of the data set *Adlershof* because the objects are moving from the right end of the scene to the left end.

Moreover, the complexity of the computation increases as the length of the filters increases.

From Figs. 9.22(b) and 9.22(e) it can be concluded that for the mother wavelet DB4 the results are much worse than those of the other two mother wavelets.



(a) (b) (c)

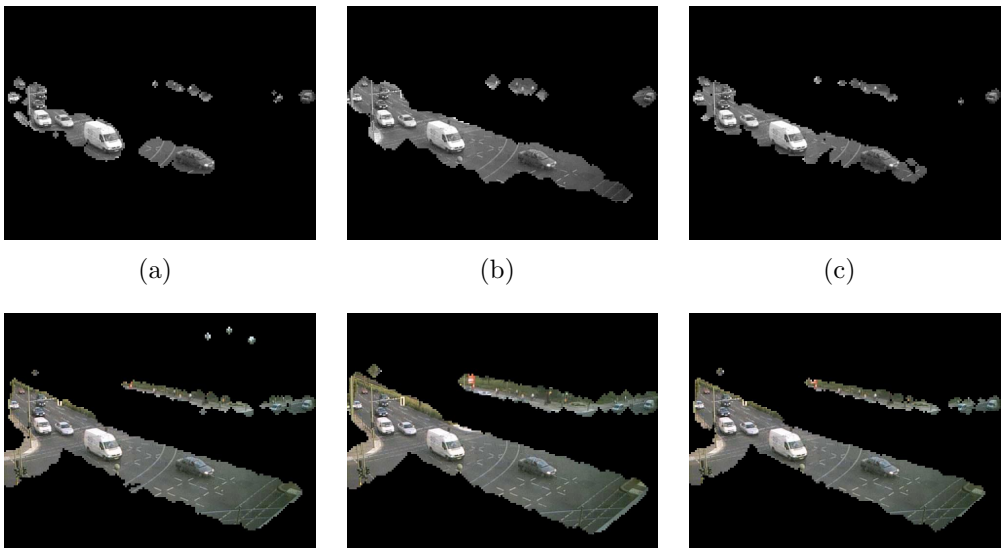
(d) (e) (f)

Figure 9.22: Extracted ROI and active traffic area for the scene *Adlershof* using the wavelets DB1, DB4, and DB8, respectively. (a), (b), and (c) Extracted ROI using the first analysis level. (d), (e), and (f) Extracted active traffic area using the third analysis level.

Table 9.15: Results of the Daubechies wavelets DB1, DB4 and DB8.

			DB1		DB4		DB8	
			No.	Percentage	No.	Percentage	No.	Percentage
Adlershof1	Level 1	*BB*	14		24		22	
		FA	2	14.3%	11	45.8%	9	40.9%
		MO	0	0.0%	0	0.0%	0	0.0%
		DD	0	0.0%	0	0.0%	0	0.0%
	Level 2	*BB*	20		36		24	
		FA	5	25.0%	25	69.4%	11	45.8%
		MO	0	0.0%	2	15.4%	0	0.0%
		DD	0	0.0%	2	15.4%	0	0.0%
Adlershof2	Level 1	*BB*	12		22		22	
		FA	1	8.3%	10	45.5%	11	50.0%
		MO	0	0.0%	0	0.0%	0	0.0%
		DD	0	0.0%	0	0.0%	0	0.0%
	Level 2	*BB*	16		36		28	
		FA	5	31.3%	24	66.7%	19	67.9%
		MO	0	0.0%	0	0.0%	0	0.0%
		DD	0	0.0%	0	0.0%	0	0.0%
	Level 3	*BB*	16		32		32	
		FA	5	31.3%	24	75.0%	21	65.6%
		MO	0	0.0%	3	27.3%	0	0.0%
		DD	0	0.0%	3	27.3%	0	0.0%
Danziger6	Level 1	*BB*	268		184		206	
		FA	1	0.4%	10	5.4%	27	13.1%
		MO	51	11.1%	64	13.1%	115	23.5%
		DD	31	6.8%	44	9.0%	99	20.2%
	Level 2	*BB*	278		148		140	
		FA	13	4.7%	29	19.6%	25	17.9%
		MO	7	1.5%	30	6.1%	61	12.4%
		DD	1	0.2%	24	4.9%	37	7.6%
	Level 3	*BB*	200		136		136	
		FA	7	3.5%	42	30.9%	42	30.9%
		MO	1	0.2%	43	8.8%	47	9.6%
		DD	1	0.2%	21	4.3%	35	7.1%
Danziger7	Level 1	*BB*	144		94		104	
		FA	5	3.5%	18	19.1%	27	26.0%
		MO	8	3.5%	20	8.1%	39	15.7%
		DD	2	0.9%	14	5.6%	33	13.3%
	Level 2	*BB*	112		108		104	
		FA	6	5.4%	43	39.8%	38	36.5%
		MO	2	0.9%	5	2.0%	11	4.4%
		DD	2	0.9%	4	1.6%	10	4.0%
	Level 3	*BB*	88		72		48	
		FA	1	1.1%	18	25.0%	8	16.7%
		MO	0	0.0%	8	3.2%	32	12.9%
		DD	0	0.0%	0	0.0%	24	9.7%

In Fig. 9.23 the extracted ROI and the active traffic areas can be found using different mother wavelets for the data set *Danziger6*. The results are obtained from the second level of the analysis. Two observations are typical. First, the results for DB4 are worse than the results of the other two wavelets. Second, using DB4 and DB8 the extracted ROI of a single group of frames are very close to the extraction of the active traffic areas of the whole sequence.



(a) (b) (c)

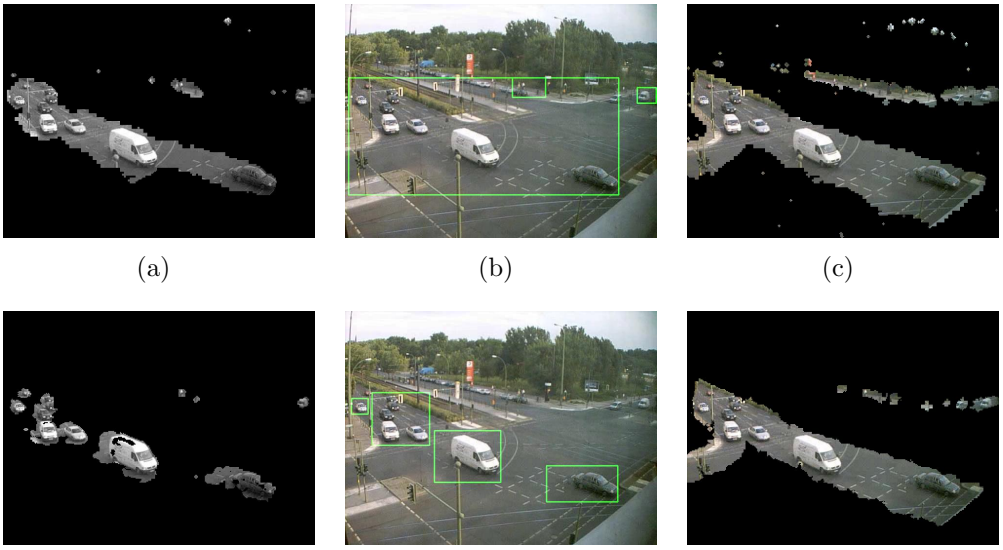
(d) (e) (f)

Figure 9.23: Extracted ROI (first row) and active traffic area (second row) for the scene *Danziger* using (a) and (d) DB1. (b) and (e) DB4. (c) and (f) DB8.

Therefore, the use of higher order Daubechies wavelets is not preferable. For the addressed application, the detection of moving objects, the use of the Haar wavelet is recommended.

9.4.4 Interresolution Masks

The discussed results so far were obtained from three different analysis levels, which are independent of each other. Here, results are presented, which were obtained from interresolution masks as introduced in Section 6.4. The masks were created using combinations of different analysis levels to increase the quality of the extraction ROI and the active traffic area.



(a) (b) (c)

(d) (e) (f)

Figure 9.24: Extracted ROI and active traffic area for the scene *Danziger* using the interresolution masks. (a), (b), and (c) OR operator. (d), (e), and (f) AND operator.

Results of ROI and active traffic area extraction are shown in Fig. 9.24 and Fig. 9.25 using different combination methods. The corresponding statistical results are displayed in Tabs. 9.16 and 9.17 for the extraction of ROI and the active traffic area, respectively.

The OR operator gives the highest rate of false alarms but lowest rates in terms of missed objects and delayed detections. The reason is that any detected movement in all levels is considered by this combination method. It inherits the failures from the first analysis level to avoid the disturbance of the small non-interesting movements and the problem of the too large ROI from the third analysis level, e.g., smallest bonding boxes in Fig. 9.24(b). This combination method can only be used when it is much more important to detect a moving object than to extract a region for further processing.

In the case of an AND operator, the false alarm rates are very low but the missed object and delayed detection rates are very high.

It can be concluded that the combination methods based on the logical operators OR and AND cannot help to improve the extraction of moving objects.

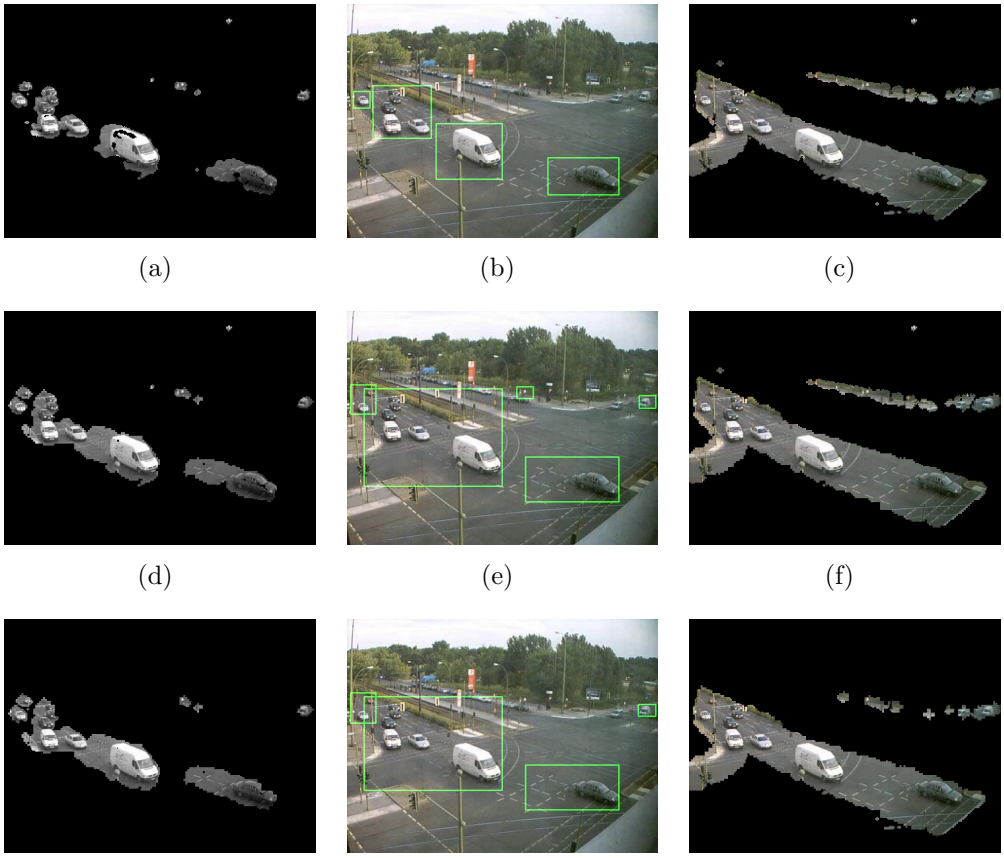


Figure 9.25: Extracted ROI and active traffic area for the scene *Danziger* using the interresolution masks. (a), (b), and (c) Third combination method. (d), (e), and (f) Fourth combination method. (g), (h), and (i) Fifth combination method.

The results of the other combination methods are found to enhance the results obtained by a single analysis level. In terms of false alarm rate they are all better compared to the results of the second and the third level of the 3D wavelet-based algorithm. In Fig. 9.25(b) one false alarm is removed compared to Fig. 9.17(b). Evaluating Tab. 9.16 the fourth combination method seems to be a good compromise between the rate of false alarms and the rate of the missed objects.

In Tab. 9.17 the precisions are given for the extraction of the active traffic area. The best results are obtained by the AND operator in contrast to the results of the extraction of the ROI (Tab. 9.16).

Table 9.16: Results of the ROI extraction using interresolution masks in terms of extracted bounding boxes.

	Method	*BB*	*FA*		*MO*		*DD*	
Adlershof	Comb1	36	12	33.3%	0	0.0%	0	0.0%
	Comb2	26	2	7.7%	0	0.0%	0	0.0%
	Comb3	12	1	8.3%	0	0.0%	0	0.0%
	Comb4	16	5	31.3%	0	0.0%	0	0.0%
	Comb5	16	5	31.3%	0	0.0%	0	0.0%
Danziger	Comb1	391	97	24.8%	4	0.5%	4	0.5%
	Comb2	388	9	2.3%	118	15.8%	103	13.8%
	Comb3	446	10	2.2%	86	11.5%	67	8.9%
	Comb4	387	26	6.7%	43	5.7%	31	4.1%
	Comb5	366	26	7.1%	55	7.3%	35	4.7%
Rudower9	Comb1	48	0	0.0%	0	0.0%	0	0.0%
	Comb2	50	0	0.0%	0	0.0%	0	0.0%
	Comb3	52	4	7.7%	0	0.0%	0	0.0%
	Comb4	52	4	7.7%	0	0.0%	0	0.0%
	Comb5	48	0	0.0%	0	0.0%	0	0.0%
Frankfurt	Comb1	450	164	36.4%	2	0.5%	0	0.0%
	Comb2	370	33	8.9%	90	24.2%	28	7.5%
	Comb3	434	43	9.9%	39	10.5%	3	0.8%
	Comb4	466	98	21.0%	32	8.6%	2	0.5%
	Comb5	412	96	23.3%	58	15.6%	27	7.3%
AdlershofAlt15	Comb1	132	8	6.1%	34	4.2%	16	2.0%
	Comb2	252	1	0.4%	279	34.1%	218	26.6%
	Comb3	286	2	0.7%	123	15.0%	64	7.8%
	Comb4	206	9	4.4%	68	8.3%	42	5.1%
	Comb5	188	9	4.8%	207	25.3%	186	22.7%
RuskaUfer	Comb1	1820	573	31.5%	94	2.8%	22	0.7%
	Comb2	2418	230	9.5%	558	16.6%	205	6.1%
	Comb3	2608	402	15.4%	360	10.7%	128	3.8%
	Comb4	2159	373	17.3%	236	7.0%	78	2.3%
	Comb5	2020	296	14.7%	373	11.1%	107	3.2%
Stuttgart	Comb1	1662	372	22.4%	246	10.9%	138	6.1%
	Comb2	1214	80	6.6%	894	39.7%	446	19.8%
	Comb3	1458	176	12.1%	740	32.8%	288	12.8%
	Comb4	1412	137	9.7%	474	21.0%	228	10.1%
	Comb5	1208	76	6.3%	661	29.3%	287	12.7%

The high missed object rates and the low false alarm rates can be explained by the density of the traffic. In this case the extracted regions are overlapping in all three analysis levels. The objects that are missed in one mask are detected later in one of the following masks. This way the active traffic area is covered well. In the same time most of the movements in the background are eliminated by the masks of high analysis levels. Hence, very few parts from the background are taken into the estimated foreground giving low over segmentation.

The lowest precisions are found in the method of the OR operator. This is due to the fact that all the extracted blobs from all levels are added together and so the areas of over segmentation are increased. The precisions of the other combination methods are between the AND and OR operators.

The second combination method seems to be suitable for the extraction of the active traffic area. The computation is simple and leads with a few frames to convenient results.

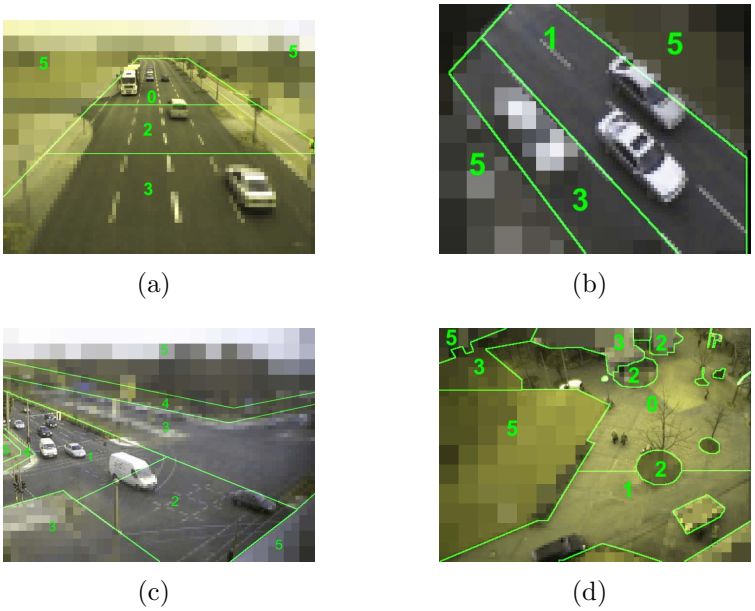
Table 9.17: Results for the extraction of active traffic area using interresolution masks in terms of the precision of over and under segmentation.

	Pr_{Comb1}	Pr_{Comb2}	Pr_{Comb3}	Pr_{Comb4}	Pr_{Comb5}
Adlershof1	84.1%	92.1%	-	-	-
Adlershof2	68.8%	89.2%	87.4%	82.1%	82.3%
Danziger6	76.1%	93.3%	86.5%	85.5%	89.7%
Danziger7	76.3%	94.3%	87.1%	86.9%	91.7%
Frankfurt10	88.5%	93.2%	93.6%	92.2%	92.0%
Frankfurt11	73.3%	87.8%	86.1%	81.0%	81.3%
RuskaUfer13	85.5%	97.4%	96.0%	95.1%	95.9%
RuskaUfer14	91.5%	96.2%	95.5%	95.1%	95.1%
AdlershofAlt15	84.5%	97.2%	96.0%	92.6%	93.1%
RuskaUfer23	77.8%	94.4%	92.7%	91.0%	92.2%
RuskaUfer24	70.0%	97.0%	92.2%	91.8%	94.5%
RuskaUfer26	65.6%	78.3%	76.0%	75.8%	77.8%
RuskaUfer27	89.0%	97.2%	95.3%	94.3%	95.9%

9.5 Results of the Resolution Mosaic Video Segmentation

In this section the results are presented and discussed obtained using the algorithm proposed in Chapter 7 for moving object detection in video surveillance. The previous results for the extraction of the active traffic area from the 3D wavelet-based algorithm are used as a basis to develop the mosaic maps. They were created manually by a human operator. For each data set the area of the background is given the lowest resolution level, while the active traffic area is divided into regions such that the close views are given low resolution levels and the far views high resolution levels.

In Fig. 9.26 some scenes are selected which are composed with different resolutions. The mosaic maps are mounted on the top of the images. The numbers mounted on the images indicate the number of the analysis levels used by the 2D wavelet transform to create lower resolution levels. The smaller the number the higher the resolution. For example, the number 5 indicates the fifth analysis level, where each block consists of 32×32 pixels and the number 0 indicates the original resolution.



(a) (b)

(c) (d)

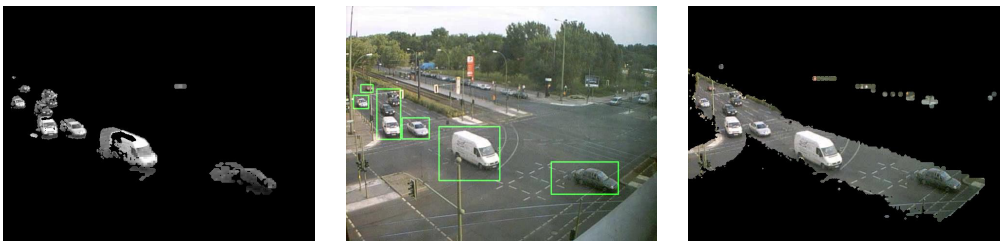
Figure 9.26: Resolution mosaic for selected scenes.

The far views in the scenes are analysed in the highest resolutions. In such far views the objects appear smaller and move slowly. The close views are analysed in low resolutions, because the objects usually appear bigger and move faster than in the other parts of the scene. The background is analysed in the lowest resolution, since no relevant information is expected from it.

As soon as the resolution mosaic of two successive frames is ready using the 2D wavelet transform, the resulting coefficients, the approximation and all details, are analysed by the 1D wavelet transform in the time domain for only one level.

Similar to the other introduced algorithms the results here are given statistically in terms of smallest bounding boxes and graphically using selected images that help in the discussion.

Fig. 9.27 gives an example of the outputs of the algorithm when applied to the data set *Danziger6*. Fig. 9.27(a) shows the ROI extraction corresponding to the frame shown in Fig. 9.6(e). Each blob in this image represents the detected motion in two successive frames. In Fig. 9.27(b) the corresponding bounding boxes are shown, in Fig. 9.27(c) the active traffic area. The results are better than those obtained so far: the bounding boxes are smaller and the results are less affected by noise and disturbances caused by unwanted movements in the background.



(a) (b) (c)

Figure 9.27: Extracted ROI and active traffic area for the scene *Danziger* using resolution mosaic 2D+1D algorithm.

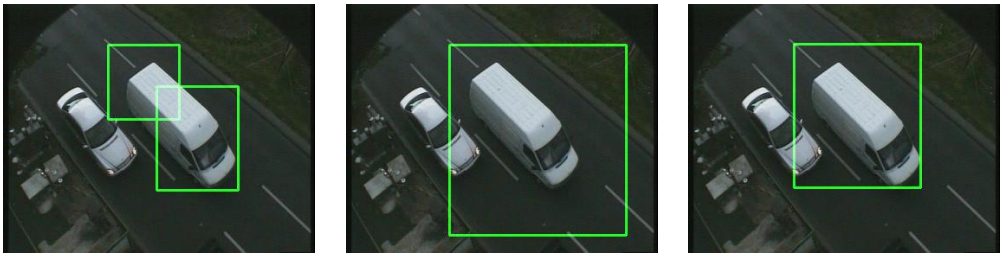
Tab. 9.18 contains the results in terms of the smallest bounding boxes. They can be compared to the results of the fourth combination method. In the data sets *Adlershof* the number of bounding boxes is decreased as well as the rate of the false alarms. The rates of missed objects and delayed detections are as before the best that can be obtained. The results of the data sets *Danziger* are only a little better.

Table 9.18: Results of extracted ROI using the resolution mosaic 2D+1D algorithm in terms of extracted bounding boxes.

	BB	FA		MO		DD	
Adlershof1	14	1	7.1%	0	0.0%	0	0.0%
Adlershof2	12	1	8.3%	0	0.0%	0	0.0%
Danziger6	346	6	1.7%	22	5.0%	22	5.0%
Danziger7	168	16	9.5%	13	5.7%	13	5.7%
Frankfurt10	110	5	4.5%	8	5.0%	1	0.6%
Frankfurt11	178	15	8.4%	25	12.8%	5	2.6%
Danziger13	244	13	5.3%	25	9.5%	11	4.2%
Danziger14	500	21	4.2%	31	5.4%	7	1.2%
Stuttgart16	488	22	4.5%	0	0.0%	0	0.0%
Stuttgart20	1302	148	11.4%	206	10.6%	84	4.3%
Danziger23	812	44	5.4%	44	3.7%	27	2.3%
Danziger24	774	227	29.3%	20	2.1%	0	0.0%
Danziger26	502	19	3.8%	52	7.7%	0	0.0%
Danziger27	982	80	8.1%	162	13.7%	33	2.8%

The data sets *Frankfurt* are much better than that obtained by the interresolution masks. The number of bounding boxes is strongly reduced and so is the number of the false alarms from 98 (obtained by the fourth combination method) to 20 (the sum of the data sets *Frankfurt10* and *Frankfurt11*). The rate of the missed objects is increased from 8.6% to 9%. In Fig. 9.28 a comparison is given between the segmentation results using the 3D wavelet-based algorithm 9.28(a), the fourth combination method of the interresolution masks 9.28(b) and the resolution mosaic 2D + 1D wavelet algorithm 9.28(c). The result of the resolution mosaic 2D +1D wavelet is better than the other two results. Due to the darkness of the scene the 3D wavelet-based algorithm has detected the moving objects in two parts, the interresolution masks gave a large bounding box around the moving object. The resolution mosaic overcomes both drawbacks.

The results of the data sets *RuskaUfer* compared to the results of the fourth combination method of the interresolution masks are much better considering all the three error measures.



(a) (b) (c)

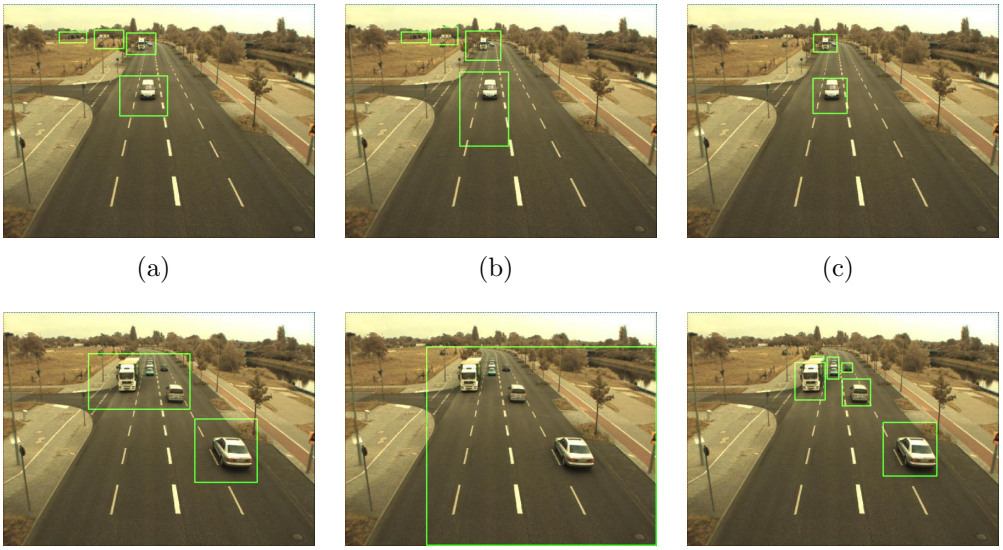
Figure 9.28: Selected results for the scene *Frankfurt* with bad lighting conditions. (a) Using the 3D wavelet-based algorithm. (b) Interresolution masks. (c) Resolution mosaic 2D+1D algorithm.

The best reduction is the false alarm rate from 17.3% to 6.1%. In Fig. 9.29 a simple comparison shall point out the advantages of the use of the resolution mosaic over the conventional 3D wavelet transform and the interresolution masks. False alarms in the background are eliminated. In the same time the detection of objects in the far view as well as in the close view are not any more too large, i.e., the new results are better in the sense of the localisation of the moving objects.

Figs. 9.29(d), 9.29(e), and 9.29(f) show another example regarding the localisation for another frame. The number of bounding boxes generated by the resolution mosaic 2D + 1D wavelet algorithm is less than that generated by the other two algorithms. The results show better localisation and more visually convenience.

The segmentation of moving pedestrians in the *Stuttgart* scenes shows that the resolution mosaic 2D + 1D wavelet algorithm outperforms almost all the other methods in terms of the missed object rate 8.8%, and delayed detection rate 3.6%. The false alarm rate 9.5% is only a little bit reduced corresponding to that obtained by the fourth combination method of the interresolution masks.

As generalized evaluation can concluded that, this method gives better results than the other studied methods without the need to have a compromise between the false alarm rates and the missed object rates.



(a) (b) (c)

(d) (e) (f)

Figure 9.29: Selected results for the scene *RuskaUfer*. (a) and (d) Using the 3D wavelet-based algorithm. (b) and (e) Interresolution masks. (c) and (f) Resolution mosaic 2D+1D algorithm.

Chapter 10

Conclusion

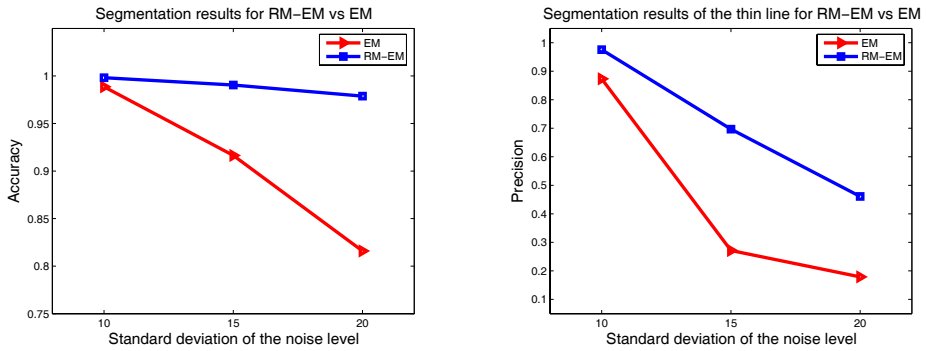
Accurate and reliable image and video segmentation is one of the challenges in image processing. Although, much work has been contributed towards this goal, there are still numerous areas for further research.

The contribution of this thesis is the development of new algorithms for image and video segmentation that are based on the multiresolution analysis. The algorithms require no prior information about the desired results. They have been evaluated against known algorithms from the literature using different criteria related to the investigated applications. The main assumption is that the multiresolution analysis simplifies and speeds up the segmentation process and increases the accuracy of the results. Various recent methods for still image segmentation as well as for video segmentation have been surveyed and are discussed according to the needs of the investigated applications.

In the theoretical background of this thesis two subject areas are the centre of attention: the multiresolution analysis and the fundamentals of the variant types of wavelet transform. It is shown that the concept of multiresolution can be regarded as independent of the wavelet transform. It is an analysis tool that is older than the wavelet transform and can be performed using different techniques. The main topic of the wavelet transform fundamentals is the expansion of the basic concept of the transform in two and three dimensions. Various Daubechies wavelets have been introduced and are discussed. It has been shown that the Haar wavelet is the best wavelet for the proposed algorithms.

For still image segmentation the Resolution Mosaic Expectation Maximization algorithm (RM-EM) is proposed. The image is pre-processed in such a way, that it is represented in a mosaic of different resolutions. The level of the resolution for a certain part of the image is based on the information content of this part. The new representation is saved in a list. For the segmentation the conventional EM algorithm is applied to this list.

Fig. 10.1 shows a comparison between the results of the conventional EM and the RM-EM algorithms for the synthetic quadratic images. The conventional EM algorithm is more sensitive to the noise level than the RM-EM algorithm. As the noise level increased from std = 10 to std = 20, the overall accuracy of the EM algorithm dropped from 99% to 82%. The accuracy of the RM-EM remained nearly constant for the same increase of the noise level. Even for the thin line class the RM-EM algorithm gave notable better precision compared to the EM algorithm.



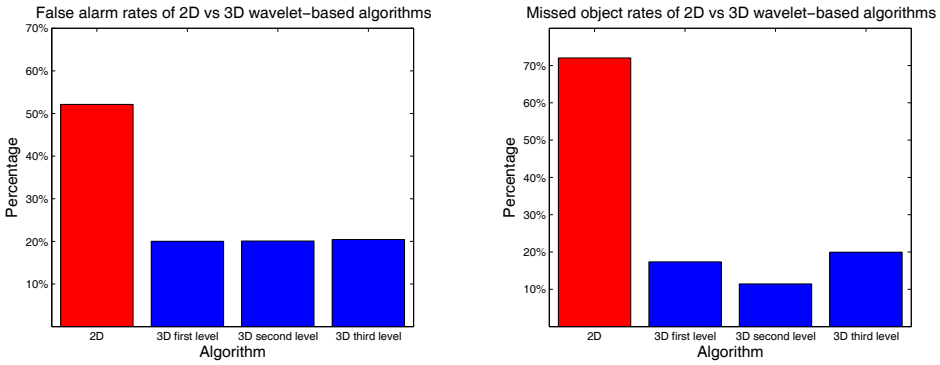
(a) (b)

Figure 10.1: Sensitivity to noise of the conventional EM and the resolution mosaic EM. (a) Overall accuracy. (b) Precision of the thin class.

The segmentation process is significantly speeded up. The number of iterations needed by the algorithm is reduced from 737 to 25 when a real MRI is segmented by the RM-EM instead of the EM.

For image sequence segmentation the 3D wavelet-based algorithm is proposed. The algorithm is able to detect moving objects, e.g., in traffic monitoring systems. The most widely used method for such task is based on background subtraction.

The use of the 3D wavelet transform avoids the difficulties of the background estimation and updating in the background subtraction-based algorithms. The 3D wavelet-based algorithm is able to detect the simultaneous spatial and temporal changes. In the investigated application the detection of moving objects means to find the moving cars in the observed scene. To accomplish this, regions of interest are extracted using the 3D wavelet transform as a primary segmentation step. Then the segmentation is improved by conventional procedures. Finally, the interesting regions are extracted from the original image sequence by a projection step. This can be considered as an advantage of the multiresolution analysis, where the processing is done on a compressed version of the data while the extracted areas are in the resolution of the original sequence. The results in terms of false alarm rates and missed object rates are much improved compared to the 2D wavelet-based algorithm. Fig. 10.2 shows two bar charts comparing the results summarised from all sets.



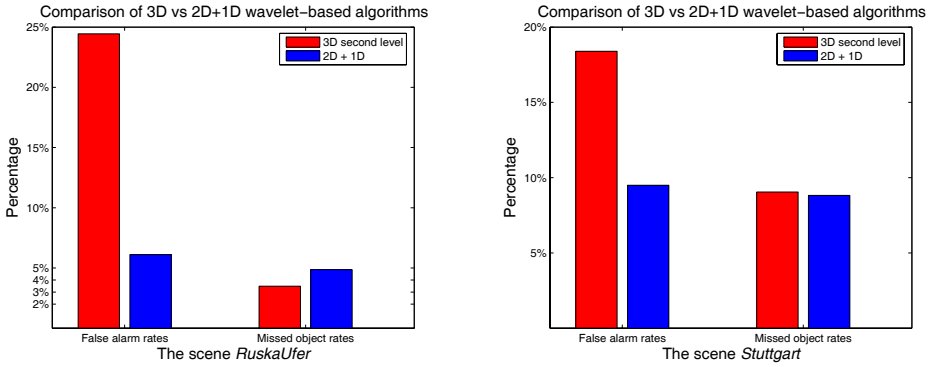
(a) (b)

Figure 10.2: Comparison between the 2D and 3D wavelet-based algorithms. (a) Rate of false alarms. (b) Rate of missed objects.

Different mother wavelets from the Daubechies family have been tested. The increasing length of the filters associated with the higher order wavelets leads to overlapping in the analysis. Due to this overlapping the sharp edges are detected as wide events. For moving object detection in image sequences, the localisation of the detection plays an important role. Therefore, the overlapping analysis has to be considered as a disadvantage. It affects the segmentation by producing enlarged areas.

Therefore, the use of the higher order Daubechies wavelets cannot be recommended. For the addressed application of the moving object detection the use of the Haar wavelet is preferable.

In other applications, such as lossy video compression, the overlapping analysis can be counted as an advantage. The information content in a certain segment of the video will be copied in various coefficients. Discarding some of the coefficients for the purpose of compression will not lead to a loss of information, since they are represented somewhere else. Thus, as the overlapping increases, high compression rates with better reconstruction of the video can be achieved.



(a) (b)

Figure 10.3: Comparison between the 3D and the 2D + 1D wavelet-based algorithms for the scenes (a) RuskaUfer. (b) Stuttgart.

The second proposal for moving object detection in image sequences is the use of the resolution mosaic and the 3D wavelet packet analysis. The new algorithm is based on analysing the spatial domain independently on the temporal domain. This way it was possible to overcome the first drawback of the 3D wavelet-based segmentation, namely the enforcement to process the image sequence in the same spatial and temporal resolution. The 2D + 1D algorithm uses a proper resolution for each part of the scene. The principle is to disregard the noise that may be present in the background, and to handle the motion information in resolutions adjusted to the features of the motion such as speed and size of the moving objects.

The results of the experiments were much better than those obtained by the 3D wavelet-based algorithm considering all tested scenes. For some special scenes such as the scene of the Stuttgart university campus, the new algorithm gave acceptable results considering all error measures as shown in Fig. 10.3.

It is not yet possible to create the mosaic map automatically. Based on the visual information of the scene and the motion parameters it could be possible to generate dynamic mosaic maps without prior information or interactive processing.

We have also proposed a concept for a hardware implementation of the 3D wavelet transform. A part of the 3D wavelet-based algorithm was implemented on a Virtex-II Pro FPGA as a primary segmentation step. For image acquisition the FPGA was connected to a digital camera using 100-Mbit Ethernet. The FPGA design was able to transform the acquired images as well as the results of the primary segmentation. The rate of 25 fps (PAL) or 30 fps (NTSC) is achieved, which is the acquisition rate of the digital camera. A client PC then processes the results of the primary segmentation further. The system is considered as an embedded vision system which uses fast hardware for data acquisition and processing.

Although only a few applications have been tested, the proposed algorithms can be applied in a wide range of other applications. We may test the 3D wavelet-based algorithm for dynamic segmentation of multichannel electroencephalographic (EEG) brain signals. An automatic detection of relevant temporal and spatial changes in the EEG map or map sequence could help for the efficient analysis of various brain activity states, e.g., the reaction to stimulation.

An open direction for future work could be to extend the use of multiresolution analysis to fields of the high-level image processing, such as content based-image processing and video indexing.

The wide spread of amateur multimedia data on news websites and in online social networks suggest to the computer science community to develop algorithms for efficient access, storage, and manipulation of the data using semantic features. The amateur multimedia data in general suffer from the lack of supporting meta data such as meaningful names or keywords in the description.

Methods that are based on keywords for indexing and querying are not any more fitting the available data. New indexing methods based on scene classification, motion information, or object recognition are nowadays highly demanded. For example, in sport videos it may be needed to extract automatically short shots that contain certain activities: offensive-defensive activities, events: foul and goal, or objects: player or commercials.

Finally, it is noted, that designing an artificial vision system with similar performance as the human vision system is a multidisciplinary challenge. Contributions from the fields of computer vision, artificial intelligence, biomedical engineering, and psychology are needed. The proposed concepts and algorithms in this thesis are hoped to make a contribution to future developments.

Bibliography

[AAB+84] Adelson, Edward H.; Anderson, C. H.; Bergen, J. R.; Burt, Peter J.; Ogden, J. M.: Pyramid methods in image processing. In: *RCA Engineer*, volume 29(6), 1984.

[AB94] Adams, Rolf; Bischof, Leanne: Seeded region growing. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (tPAMI)*, volume 16:pp. 641–647, 1994.

[ACHTSN06] Antón-Canalís, Luis; Hernández-Tejera, Mario; Sánchez-Nielsen, Elena: Addcanny: Edge detector for video processing. In: *ACIVS*, pp. 501–512. 2006.

[AJN97] Alirezaie, J.; Jernigan, M.E.; Nahmias, C.: Neural network-based segmentation of magnetic resonance images of the brain. In: *IEEE Transactions on Nuclear Science*, volume 44(2):pp. 194–198, April 1997.

[ARS01] ARS Traffic and Transport Technology BV, Vlietweg 14, 2266 KA, Leidschendam, The Netherlands: *Traffic Monitoring and Analysis*, v.1.2 edition, October 2001.

[AU96] Aldroubi, A.; Unser, M.A., editors: *Wavelets in Medicine and Biology*. CRC Press, Boca Raton FL, USA, 1996.

[AZ08] Allili, Mohand Said; Ziou, Djemel: Object tracking in videos using adaptive mixture models and active contours. In: *Neurocomputing In Press*, 2008.

[BBRS04] Bramberger, M.; Brunner, J.; Rinner, B.; Schwabach, H.: Real time video analysis on a smart camera for traffic surveillance. In: *10th IEEE Real-Time and Imbedded Technology and Applications Symposium (RTAS 04)*. IEEE Computer Society, 2004.

[BC77] Bezdek, James C.; Castelaz, Patrick F.: Prototype classification and feature selection with fuzzy sets. In: *IEEE Transactions on Systems, Man and Cybernetics*, volume 7(2):pp. 87–92, 2 1977.

[Bey92] Beylkin, G.: On the representation of operators in bases of compactly supported wavelets. In: *SIAM J. Numer. Anal.*, volume 29(6):pp. 1716–1740, 1992.

[Bla98] Blatter, Christian: *Wavelets: A Primer*. A K Peters, Ltd., 1998.

[BMCM97] Beymer, D.; McLauchlan, P.; Coifman, B.; Malik, J.: A real-time computer vision system for measuring traffic parameters. In: *Computer Vision and Pattern Recognition*, pp. 495 – 501. IEEE Computer Society, California Univ., Berkeley, CA, USA, June 17-19 1997.

[Bon97] Bonet, Jeremy S. De: Multiresolution sampling procedure for analysis and synthesis of texture images. In: *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 361–368. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997.

[Bov00] Bovik, A.C.: *Handbook of Image and Video Processing*. Academic Press, May 2000.

[BP93] Brunelli, R.; Poggio, T.: Face recognition: Features versus templates. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15(10):pp. 1042–1052, 1993.

[Bur84] Burt, P. J.: The pyramid as a structure of efficient computation. In: *Multiresolution Image Processing and Analysis*, pp. 6–35. Springer-Verlag, 1984.

[Bur98] Burges, Christopher J. C.: A tutorial on support vector machines for pattern recognition. In: *Data Mining and Knowledge Discovery*, volume 2(2):pp. 121–167, 1998.

[BW06] Brannock, Evelyn; Weeks, Michael: Edge detection using wavelets. In: *ACM-SE 44: Proceedings of the 44th Annual Southeast Regional Conference*, pp. 649–654. ACM, New York, NY, USA, 2006.

[Can86] Canny, J.: A computational approach to edge detection. In: *IEEE Transaction Pattern Analysis and Machine Intelligence*, volume 8(6):pp. 679–698, November 1986.

[Cas96] Castleman, Kenneth R.: *Digital Image Processing.* Prentice Hall, Inc., 1996.

[CD00] Comer, M.L.; Delp, E.J.: The EM/MPM algorithm for segmentation of textured images: Analysis and further experimental results. In: *IEEE Transactions on Image Processing*, volume 9(10):pp. 1731–1744, October 2000.

[CGE02] Cavallaro, A.; Gelasca, E.D.; Ebrahimi, T.: Objective evaluation of segmentation quality using spatio-temporal context. In: *ICIP02*, pp. III: 301–304. 2002.

[CJDC02] Chateau, Thierry; Jurie, Frédéric; Dhome, Michel; Clady, Xavier: Real-time tracking using wavelet representation. In: *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, pp. 523–530. Springer-Verlag, London, UK, 2002.

[CL94] Chang, Yian-Leng; Li, Xiaobo: Adaptive image region growing. In: *IEEE Transactions on Image Processing*, volume 3:pp. 868–872, November 1994.

[CLK+00] Collins, R.T.; Lipton, A.J.; Kanade, T.; Fujiyoshi, H.; Duggins, D.; Tsin, Y.; Tolliver, D.; Enomoto, N.; Hasegawa, O.; Burt, P.; Wixson, L.: A system for video surveillance and monitoring. Technical Report CMURI-TR-00-12, Carnegie Mellon University, 2000.

[CLL02] Chen, Li-Fen; Liao, Hong-Yuan Mark; Lin, Ja-Chen: Wavelet-based optical flow estimation. In: *IEEE Trans. Circuits Syst. Video Techn.*, volume 12(1):pp. 1–12, 2002.

[CMQW93] Coifman, Ronald Raphael; Meyer, Yves; Quake, Stephen R.; Wickerhauser, Mladen Victor: Signal processing and compression with wavelet packets. In: Meyer, Yves; Roques, Sylvie, editors, *Progress in Wavelet Analysis and Applications*, Proceedings of the International Conference "Wavelets and Applications," Toulouse, France, 8–13 June 1992, pp. 77–93. Observatoire Midi-Pyrénées de l'Université Paul Sabatier, Editions Frontieres, Gif-sur-Yvette, France, 1993.

[CS05] Chan, Tony; Shen, Jianhong: *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.

[CW04] Chen, Yixin; Wang, James Z.: Image categorization by learning and reasoning with regions. In: *J. Mach. Learn. Res.*, volume 5:pp. 913–939, 2004.

[Dau92] Daubechies, Ingrid: *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[DB00] Demuth, Howard; Beale, Mark: *Neural Network Toolbox User's Guide*. The Mathworks. Inc., 2000.

[Dim02] Dima, Anca: *Computer Aided Image Segmentation and Graph Construction of Nerve Cells from 3D Confocal Microscopy Scan*. Ph.D. thesis, Technical University Berlin, Department of Neuronal Information Processing, 2002.

[DLR97] Dempster, A. P.; Laird, N. M.; Rubin, D. B.: Maximum likelihood from incomplete data via the EM algorithm. In: *Journal of Royal Statistics Society*, volume 38-B:pp. 1–38, 1997.

[DM01] Deng, Yining; Manjunath, B. S.: Unsupervised segmentation of color-texture regions in images and video. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, volume 23(8):pp. 800–810, 2001.

[DMM03] Desolneux, Agnes; Moisan, Lionel; Morel, Jean-Michel: Computational gestalts and perception thresholds. In: *Journal of Physiology-Paris, Neurogeometry and Visual Perception*, volume 97(2-3):pp. 311–324, March-May 2003.

[EDS+05] Eeckhaut, Hendrik; Devos, Harald; Schrauwen, Benjamin; Christiaens, Mark; Stroobandt, Dirk: A hardware-friendly wavelet entropy codec for scalable video. In: *DATE '05: Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 14–19. IEEE Computer Society, Washington, DC, USA, 2005.

[Fas05] Fastenrath, Ulrich: *Floating Car Data on a Larger Scale*. DDG Gesellschaft für Verkehrsdaten mbH, Niederkasseler Lohweg 20, 40547 Düsseldorf, June 2005.

[FFJ05] Feng, Yue; Fang, Hui; Jiang, Jianmin: Region growing with automatic seeding for semantic video object segmentation. In: *ICAPR (2)*, pp. 542–549. 2005.

[FR07] Fowler, James E.; Rucker, Justin T.: *Three-Dimensional Wavelet-Based Compression of Hyperspectral Imagery*, pp. 379–407. Hyperspectral Data Exploitation. John Wiley and Sons, Inc., 2007.

[FZBH05] Fan, Jianping; Zeng, Guihua; Body, Mathurin; Hacid, Mohand-Said: Seeded region growing: An extensive and comparative study. In: *Pattern Recogn. Lett.*, volume 26(8):pp. 1139–1156, 2005.

[GC99] Goswami, Jaideva C.; Chan, Andrew K.: *Fundamentals of Wavelets: Theory, Algorithms, and Applications*. A Wiley inter-science publication, 1999.

[GEKS06] Gelasca, Elisa Drelie; Ebrahimi, Touradj; Karaman, Mustafa; Sikora, Thomas: A framework for evaluating video object segmentation algorithms. In: *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, p. 198. IEEE Computer Society, Washington, DC, USA, 2006.

[Gua06] Guan, Ye-Peng: Automatic extraction of lip based on wavelet edge detection. In: *SYNASC*, pp. 125–132. 2006.

[GW93] Gonzalaz, Rafael G.; Woods, Richard E.: *Digital Image Processing*. Addision-Wesley publishing Co, 1993.

[GW05] Gonzalez, Rafael C.; Woods, Richard E.: *Image Processing Toolbox, 5th Edition*. The Mathworks. Inc., 2005.

[GY95] Ghavamnia, Mohammad H.; Yang, Xue D.: Direct rendering of laplacian pyramid compressed volume data. In: *VIS '95: Proceedings of the 6th Conference on Visualization '95*, p. 192. IEEE Computer Society, Washington, DC, USA, 1995.

[GYB02] Goldman, D.; Yang, M.; Bourbakis, N.: A neural network-based segmentation tool for color images. In: *ICTAI*, volume 00:p. 500, 2002.

[Hab88] Haberäcker, Peter: *Digitale Bildverarbeitung Grundlagen und Anwendungen*. Carl Hanser Verlag, München, 1988.

[HCL03] Hsu, C. W.; Chang, C. C.; Lin, C. J.: A practical guide to support vector classification. Technical report, National Taiwan University, Taipei, 2003.

[HK98] Hojjatoleslami, S. A.; Kittler, J.: Region growing: A new approach. In: *IEEE Transactions on Image Processing*, volume 7:pp. 1079–1084, 1998.

[HL07] Huang, Zhi-Kai; Liu, De-Hui: Segmentation of color image using EM algorithm in HSV color space. In: *Information Acquisition, 2007. ICIA '07. International Conference on*, pp. 316–319. Jeju, Korea, July 2007.

[HS92] Horn, Berthold K. P.; Schunck, Brian G.: Determining optical flow. In: , pp. 389–407, 1992.

[IP99] Ihm, Insung; Park, Sanghun: Wavelet-based 3D compression scheme for interactive visualization of very large volume data. In: Duke, David; Coquillart, Sabine; Howard, Toby, editors, *Computer Graphics Forum*, volume 18(1), pp. 3–15. 1999.

[IVs03] IVsource.net and Richard Bishop Consulting (RBC): *Floating Car Data: Methods are Gaining Momentum Worldwide*, November 2003.

[JBM+00] Jiang, X.; Bowyer, K.W.; Morioka, Y.; Hiura, S.; Sato, K.; Inokuchi, S.; Bock, M.; Guerra, C.; Loke, R.E.; du Buf, J.M.H.: Some further results of experimental comparison of range image segmentation algorithms. In: *ICPR00*, pp. Vol IV: 877–881. 2000.

[Kai98] Kaiser, Gerald: The fast Haar transform, gateway to wavelets. In: *IEEE potentials*, AprilMay 1998.

[KDM06] Knauer, U.; Dammeier, T.; Meffert, B.: The structure of road traffic scenes as revealed by unsupervised analysis of the time averaged optical flow. In: *17th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering*. Weimar, Germany, 2006.

[KK07] Khademi, April; Krishnan, Sridhar: Shift-invariant discrete wavelet transform analysis for retinal image classification. In: *Medical and Biological Engineering and Computing*, volume 45(12):pp. 1211 – 1222, December 2007.

[KM03] Kim, Z.W.; Malik, J.: Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In: *IEEE International Conference on Computer Vision*. IEEE Computer Society, 2003.

[KPZC06] Kwon, Oh-Sang; Pizlo, Zygmunt; Zelaznik, Howard N.; Chiu, George: Multi-resolution Model of Human Motor Control. In: *J. Vis.*, volume 6(6):pp. 936–936, 6 2006.

[KT05] Khare, Ashish; Tiwary, Uma Shanker: Soft-thresholding for denoising of medical images - a multiresolution approach. In: *International Journal of Wavelets, Multiresolution and Information Processing*, volume 3(4):pp. 477–496, April 2005.

[KWSW97] Kaufhold, J.; Willsky, A.; Schneider, M.; W.C.Karl: MR image segmentation and data fusion using statistical approach. In: *IEEE International Conference On Image Processing, ICIP'97*. IEEE Computer Society, October 26-29 1997.

[KZK03] Kastrinaki, V.; Zervakis, M.; Kalaitzakis, K.: A survey of video processing techniques for traffic applications. In: *Image and Vision Computing*, volume 21(4), April 2003.

[LFK08] Lee, Tong Hau; Fauzi, Mohammad Faizal Ahmad; Komiya, Ryoichi: Segmentation of CT head images. In: *BMEI '08: Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics*, pp. 233–237. IEEE Computer Society, Washington, DC, USA, 2008.

[LK81] Lucas, B. D.; Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *IJCAI81*, pp. 674–679. 1981.

[LKK03] Lee, Wangheon; Kim, Dongsu; Kweon, Inso: Automatic edge detection method for the mobile robot application. In: *Intelligent Robots and Systems (IROS03)*, volume 3, pp. 2730– 2735. 2003.

[LOPR97] Lehmann, Thomas; Oberschelp, Walter; Pelikan, Erich; Repges, Rudolf: *Bildverarbeitung für die Medizin*. Springer-Verlag Berlin, 1997.

[Lun00] Lundberg, Frans: Maximum entropy matching: An approach to fast template matching. Report LiTH-ISY-R-2313, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, October 2000.

[MAF02] Mohamed, Nevin A.; Ahmed, M.A.; Farag, A.: Modified fuzzy C-mean in medical image segmentation. In: *IEEE Transactions on Medical Imaging*, volume 21, March 2002.

[Maj08] Majumder, Aditi: Visual perception. Lectures Notes, Spring 2008.

[Mal89] Mallat, Stéphane G.: A theory for multiresolution signal decomposition, the wavelet representation. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence*, volume 2(7):pp. 674–693, 1989.

[MB95] Mortensen, Eric N.; Barrett, William A.: Intelligent scissors for image composition. In: *SIGGRAPH '95: Proceedings of The 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 191–198. ACM, New York, NY, USA, 1995.

[MBK+05] Meffert, B.; Blaschek, R.; Knauer, U.; Reulke, R.; Winkler, F.; Schischmanow, A: Monitoring traffic by optical sensors. In: *Second International Conference on Intelligent Computing and Information Systems*. Cairo, Egypt, March 5-7 2005.

[MCE04] Moses, Alan M.; Chiang, Derek Y.; Eisen, Michael B.: Phylogenetic motif detection by expectation-maximization on evolutionary mixtures. In: *Pacific Symposium on Biocomputing*, pp. 324–335. 2004.

[Met53] Metzger, Wolfgang: *Gesetze des Sehens*. Waldemar Kramer Verlag, Frankfurt am Main, 2nd edition, 1953.

[MMOP07] Misiti, Michel; Misiti, Yves; Oppenheim, Georges; Poggi, Jean-Michel: *Wavelet Toolbox User's Guide, 5th Edition*. The Mathworks. Inc., 2007.

[MNG07] Mekhalfa, F.; Nacereddine, N.; Goumeidane, A.B.: Unsupervised algorithm for radiographic image segmentation based on the gaussian mixture model. In: *EUROCON, 2007. The International Conference on "Computer as a Tool"*, pp. 289–293, September 2007.

[MQG+01] Moyano, E.; Quiles, F.J.; Garrido, A.; Duato, J.; Orozco-Barbosa, L.: Efficient 3D wavelet transform decomposition for video compression. In: *DCV '01: Proceedings of the Second International Workshop on Digital and Computational Video*, p. 118. IEEE Computer Society, Los Alamitos, CA, USA, 2001.

[NH04] Neoh, H.; Hazanchuk, A.: Adaptive edge detection for real-time video processing using FPGAs. In: *Global Signal Processing*. 2004.

[OH03] Ouerhani, Nabil; Hügli, Heinz: Maps: Multiscale attention-based presegmentation of color images. In: *Scale-Space*, pp. 537–549. 2003.

[Ols02] Olson, Clark F.: Maximum-likelihood image matching. In: *IEEE Trans. Pattern Anal. Mach. Intell.*, volume 24(6):pp. 853–857, 2002.

[Ots79] Otsu, N.: A threshold selection method from grey-level histograms. In: *SMC*, volume 9(1):pp. 62–66, January 1979.

[Par01] Park, Y.: Shape-resolving local thresholding for object detection. In: *Pattern Recognition Letters*, volume 22:pp. 883–890, 2001.

[PM87] Perona, P.; Malik, J.: Scale space and edge detection using anisotropic diffusion. In: *CVWS87*, pp. 16–22. 1987.

[Poh04] Pohle, Regina: *Computerunterstützte Bildanalyse zur Auswertung medizinischer Bilddaten*. Ph.D. thesis, Fakultät für Informatik der Otto-von-Guericke-Universität Magdeburg, 2004.

[PP93] Pal, N.R.; Pal, S.K.: A review on image segmentation techniques. In: *PR*, volume 26(9):pp. 1277–1294, September 1993.

[PR88] Pal, Sankar K.; Rosenfeld, Azriel: Image enhancement and thresholding by optimization of fuzzy compactness. In: *Pattern Recognition Letter*, volume 7(2):pp. 77–86, 1988.

[Pre84] Preston, K., Jr.: Multiresolution microscopy. In: *Multiresolution Image Processing and Analysis*, pp. 356–364. Springer-Verlag, 1984.

[RB00] Rao, Raghuver M.; Bopardikar, Ajit S.: *Wavelet Transform, Introduction to Theory and Applications.* Addison Wesley, Pearson Education, 2000.

[Rho07] Rhody, Harvey E.: Digital image processing. Lectures Note, Electrical Engineering, Syracuse University, 2007.

[Röm07] Römisch, Werner: Wavelets. Lectures Notes, Summer Semester 2007, 2007.

[Ros84a] Rosenfeld, A.: Some useful properties of pyramids. In: *Multiresolution Image Processing and Analysis*, pp. 2–5. Springer-Verlag, 1984.

[Ros84b] Rosenfeld, Azriel, editor: *Multiresolution Image Processing and Analysis.* Springer, Berlin, 1984.

[Sae97] Saeed, Mohammed: *Maximum Likelihood Parameter Estimation of Mixture Models and Its Application to Image Segmentation and Restoration.* Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1997.

[SAHU04] Sühling, Michael; Arigovindan, Muthuvel; Hunziker, Patrick R.; Unser, Michael: Multiresolution moment filters: Theory and applications. In: *IEEE Transactions on Image Processing*, volume 13(4):pp. 484–495, 2004.

[SAWM08] Salem, Mohammed A-Megeed; Appel, Markus; Winkler, Frank; Meffert, Beate: FPGA-based smart camera for 3D wavelet-based image segmentation. In: *2nd ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC-08).* Stanford University, Stanford, California, USA, September 7-11 2008.

[SD04] Singh, Jasvinder; Dana, Kristin J.: Clustering and blending for texture synthesis. In: *Pattern Recogn. Lett.*, volume 25(6):pp. 619–629, 2004.

[SG99] Stauffer, C.; Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: *CVPR99*, pp. II: 246–252. 1999.

[SMTG01] Salem, Mohammed A-Megeed; Mostafa, M. G.; Tolba, M. F.; Gharib, T. F.: Medical image segmentation using wavelet-based multiresolution EM algorithm. In: *Proceedings of Industrial Electronics, Technology and Automation IETA01*. Cairo, Egypt, 2001.

[SRKN98] Saeed, Mohammed; Rabiee, Hamid R.; Karl, W.C.; Nguyen, T.Q.: A new multi-resolution algorithm for image segmentation. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP98*, volume 5, pp. 2753 – 2756. IEEE Computer Society, May 12-15 1998.

[ST94] Shi, Jianbo; Tomasi, Carlo: Good features to track. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*. Seattle, June 1994.

[STMG03] Salem, Mohammed A-Megeed; Tolba, M. F.; Mostafa, M. G.; Gharib, T. F.: MR-brain image segmentation using gaussian multiresolution analysis and the EM algorithm. In: *ACM-IEEE 8th International Conference Of Enterprise Intelligent Computing*, volume 2, pp. 165–170. France, April 23-26 2003.

[STS07] Saleem; Touqir; Siddiqui: Novel edge detection. In: *ITNG*, pp. 175–180. IEEE Computer Society, Los Alamitos, CA, USA, 2007.

[SVMJ95] Schoner, Brian; Villasenor, John D.; Molloy, Steve; Jain, Rajeev: Techniques for FPGA implementation of video compression systems. In: *Proceedings of the 1995 ACM Third International Symposium on Field Programmable Gate Arrays,*, pp. 154–159. Association for Computing Machinery, Monterey, California, USA, February 1995.

[TCAA05] Töreyin, B. Ugur; Cetin, A. Enis; Aksay, Anil; Akhan, M. Bilgay: Moving object detection in wavelet compressed video. In: *Signal Processing: Image Communication*, volume 20:pp. 255–264, March 2005.

[Tiz98] Tizhoosh, Hamid R.: *Fuzzy-Bildverarbeitung: Einführung in Theorie und Praxis*. Springer-Verlag, 1998.

[TLE02] Tonder, Gert J. Van; Lyons, Michael J.; Ejima, Yoshimichi: Visual perception in karesansui gardens. In: *17th Congress of the International Association of Empirical Aesthetics*. Takarazuka, Japan, August 4-8 2002.

[TM96] Tood, A.; Moon, K.: The expectation maximization algorithm. In: *IEEE Signal Processing Magazine*, pp. 47–68, November 1996.

[Toe05] Toennis, Klaus D.: *Grundlagen der Bildverarbeitung*. Pearson Studium, 2005.

[TSNEA05] Tahoun, Mohamed A.; Salem, Mohammed A-Megeed; Nagaty, Khaled A.; El-Arief, Taha I.: A robust content-based image retrieval system using multiple features representations. In: *IEEE International Conference on Networking, Sensing and Control(ICNS'05)*. Arizona, USA, March 19-22 2005.

[TT07] Tang, Feng; Tao, Hai: Fast multi-scale template matching using binary features. In: *Eighth IEEE Workshop on Applications of Computer Vision (WACV'07)*, volume 0:p. 36, 2007.

[UA96] Unser, M.; Aldroubi, A.: A review of wavelets in biomedical applications. In: *Proceedings of the IEEE*, volume 84(4):pp. 626–638, April 1996.

[Vap98] Vapnik, Vladimir N.: *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[VMS99] Villegas, P.; Marichal, X.; Salcedo, A.: Objective evaluation of segmentation masks in video sequences. In: *WIAMIS 99 workshop*, pp. 85–88. Berlin, 1999.

[WA94] Wang, J.Y.A.; Adelson, E.H.: Representing moving images with layers. In: *Image Processing*, volume 3(5):pp. 625–638, September 1994.

[WNL01] Won, Y.; Nam, J.; Lee, B.-H.: Image pattern recognition in natural environment using morphological feature extraction. In: *SSPR & SPR 2000; F.J. Ferri (Ed.)*, pp. 806–815. Springer, Berlin, 2001.

[Xil05] Xilinx, Inc.: *Xilinx U069 XUP Virtex-II Pro Development System, Hardware Reference Manual*, 1st edition, March 2005.

[Xil06] Xilinx, Inc.: *Xilinx UG253 Multi Port Memory Controller 2 (MPMC2), User Guide*, 1st edition, October 2006.

[Yam98] Yamazaki, T.: Introduction of EM algorithm into color image segmentation. In: *2nd IEEE Conference on Intelligent Processing Systems, ICIPS*. Gold Coast, QLD, Australia, August, 4-7 1998.

[YWC05] Yang, Fuzheng; Wan, Shuai; Chang, Yilin: Improved method for gradient-threshold edge detector based on HVS. In: *CIS (1)*, pp. 1051–1056. 2005.

[YYK03] Yoneyama, Akio; Yeh, Chia H.; Kuo, C.-C. Jay: Moving cast shadow elimination for robust vehicle extraction based on 2D joint vehicle/shadow models. In: *AVSS '03: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, p. 229. IEEE Computer Society, Washington, DC, USA, 2003.

[ZHT05] Zou, An-Min; Hou, Zeng-Guang; Tan, Min: Support vector machines (SVM) for color image segmentation with applications to mobile robot localization problems. In: *Advances in Intelligent Computing*, volume 3645 of *Lecture Notes in Computer Science*, pp. 443–452. Springer Berlin / Heidelberg, International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August23-26 2005.

[ZK03a] Zang, Q.; Klette, R.: Evaluation of an adaptive composite gaussian model in video surveillance. In: *Computer Analysis of Images and Patterns*, pp. 165–172. 2003.

[ZK03b] Zang, Q.; Klette, R.: Object classification and tracking in video surveillance. In: *10th International Conference of Computer Analysis of Images and Patterns, CAIP*, volume 2756, pp. 198–205. Springer Berlin / Heidelberg, Groningen, The Netherlands, August 25-27 2003.

[ZN01] Zhao, T.; Nevatia, R.: Car detection in low resolution aerial image. In: *IEEE International Conference on Computer Vision*, pp. 710–717. 2001.

[ZZWF06] Zhao, Shuguang; Zhao, Jun; Wang, Yuan; Fu, Xinlin: Moving object detecting using gradient information, three-frame-differencing and connectivity testing. In: *Australian Conference on Artificial Intelligence*, pp. 510–518. 2006.

Acknowledgement

A PhD is a unique experience. Not because one does it, at most once in a life, but because it is an extremely enriching experience. During the time I spent and through this work I had the opportunity to expand my horizons in many dimensions. I have learned new research techniques, new fields, and I have strengthened my knowledge of the fundamentals. Moreover, I have learned new way of life, culture, and mentality. These all was not possible without the wonderful people around me and without the language.

First of all, I would like to thank Prof. Beate Meffert for having welcomed me to join her group, for supervising my work from the first to the last moment, and for giving me the chance to take a part in the University-life in Berlin. She gave me every possible chance to represent my work and myself and to talk in the name of the group. She helped me not only to do good research but also to gather very valuable academic experience.

In my first visit to the work group in Berlin I recognised that it is a big family. They accepted me fast and every member of this big family has helped me by his way and has added to my benefits from being in Berlin. I would like to thank Dr. Hochmuth and Dr. Winkler. I have learned from one to use regulatory procedures in everything. From the other I have learned to think simple and challenging. Actually, simple ideas are more challenging. Also I would like to thank my colleagues Mr. Blaschek, Mr. Mankiewicz, Mr. Knauer, and Mr. Heese. I would like to thank Prof. Harmuth for reviewing the English of the thesis.

A great part of my benefits and enjoyment in this phase of my life goes back to the language. Therefore, I would like to thank the teachers in Goethe-Institut in Göttingen. The language was the key that opened the doors of the country and the hearts of the people to me. My thanks go too to Ms. Leopold from DAAD for her kind supporting services.

Lebenslauf

Mein Lebenslauf wird aus Datenschutzgründen in der elektronischen Version meiner Arbeit nicht mit veröffentlicht.

Selbstständigkeitserklärung

Ich erklr̈e hiermit, dass

ich die vorliegende Dissertationsschrift selbstständig und ohne unerlaubte Hilfe angefertigt habe,

ich mich nicht bereits anderwärts um einen Doktorgrad beworben habe oder einen solchen besitze, und

mir die Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät II der Humboldt-Universität zu Berlin bekannt ist, gemäß Amtliches Mitteilungsblatt Nr. 34/2006.

Mohammed Abdel-Megeed M. Salem
Berlin, den 26. August 2008